Universiteit Leiden
The Netherlands

BSc Data Science & AI

How effective are simple machine learning models in accurately predicting structural properties of 2D perovskites directly from crystal structure data?

Dani Jonas

Supervisors:
Dr. F. Bariatti & Dr. A. Knobbe & Dr. F. Grozema

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

15/01/2026

Abstract

Despite solar cells being a sustainable way of generating energy, efficient solar cells remain challenging because materials must withstand varying weather conditions. Two-dimensional (2D) perovskites are layered crystal structures with tunable chemical properties. Since 2D perovskites have tunable properties, they are a good fit for creating solar cells, although determining which 2D perovskites are most suitable is difficult. As experimentally creating solar cells is too costly, Molecular Dynamics simulations can be used to analyse 2D perovskites, although running these simulations can be computationally expensive. Simple Machine Learning models could predict 2D perovskite properties without requiring all simulations to be performed.

In this thesis, an automated pipeline converts 2D perovskite files to simulation-ready files, after which MD simulations are performed to generate a dataset to predict properties of the 2D perovskites. The selected properties are the average potential-energy temperature derivative, and the average volume per atom. Using Least Squares Linear Regression, LASSO, Ridge and Random Forest models to predict these targets revealed that the Random Forest model consistently performed the best, and captured relevant behaviours from the provided data. These results show that simple machine learning models, especially the Random Forest model, can be used to preselect subsets of 2D perovskite materials, thereby reducing the number of required simulations.

Contents

1 Introduction

The world's demand for energy is rising due to the technological advancements [1]. Much of this energy is produced in a sustainable manner [2]. One such sustainable option is the use of solar cells, which can convert sunlight into energy. A problem, however, is the efficiency of the solar cells: creating solar cells with a high efficiency is difficult due to the solar cell panels needing to withstand varying environmental conditions and temperatures, which is dependent on the climate of the location in which the solar cells are placed. Therefore, the materials used to build solar cells should both be efficient as well as resistant. This emphasises the importance of selecting the right type of material for the construction of solar cells.

One such type of materials useful for constructing solar cells are the *perovskites*, specifically the two-dimensional (2D) perovskites. Perovskites are a class of materials identifiable by their specific crystal structure, which allows for structural variability especially present in 2D perovskites. The structural variability allows for a wide range of tunable material properties based on the composition of the material. One of the most relevant of these properties are the optoelectronic properties, which determine how efficiently a material converts light into energy [3, 4].

Despite the tunability of 2D perovskites, the structural variability also creates a challenge. There are many different possible 2D perovskite compositions, and determining which exact composition is the best for a specific use case is not feasible; experimentally creating and testing all of the compositions would be too time-consuming and costly. One possible solution would be to explore these different compositions computationally. Molecular Dynamics (MD) simulations allow for the computational simulation, and thus exploration, of different chemical compositions and structures. The MD simulations describe the interactions between atoms over time by simulating a part of the material as a box of atoms, which allows the behaviour of the material to be studied under varying conditions, such as different temperatures and pressures [5]. The problem with this approach, however, is that the simulations can be computationally expensive, making them time-consuming to perform. Therefore, a different approach is needed to analyse, and generalise, the information that can be gathered from MD simulations.

Machine learning models allow for the analysis of data, and the generalisation of predictions beyond the initial data. In the context of MD simulations, machine learning models can provide predictions of material properties without requiring every material to be simulated. Machine learning models can therefore be used to learn the relationships between the widely available structural information and the simulated material properties of the 2D perovskites.

To address these research gaps, this research aims to find an automated approach that combines the structures of the 2D perovskite materials with the MD simulations to generate a dataset containing simulated material properties of 2D perovskites. This dataset will then be analysed using machine learning models, in an attempt to create simple machine learning models that can predict certain properties of the 2D perovskites.

Manuscript Plan & Research Question

This manuscript first presents the background with regards to 2D perovskite materials and machine learning models. After this, an automated workflow is introduced, which uses structural information from the 2D perovskites and converts it into files needed to run the Molecular Dynamics simulations. The resulting dataset from these simulations contains structural information and material properties of 2D perovskites, thereby providing a basis for the analysis and construction of the machine learning models. The dataset is then used to analyse and identify suitable machine learning targets. Finally, machine learning models are trained on this data in an attempt to address the following research question: *How effective are simple machine learning models in accurately predicting the structural properties of 2D perovskites directly from structural data?*

The goal of this research is not to create machine learning models that can differentiate between all

unique 2D perovskite materials. Instead, the aim is to identify subsets of materials based on the desired properties, thereby reducing the amount of materials that require further simulation or experimental research. Since the Molecular Dynamics simulations are computationally expensive, *simple* machine learning models are used to avoid a further increase in computational resources. Moreover, simple machine learning models can provide better interpretability than complex machine learning models [6]. Interpretability is important in the context of chemistry, as it allows for a better understanding of how the structural features of the material influence the properties of the 2D perovskites.

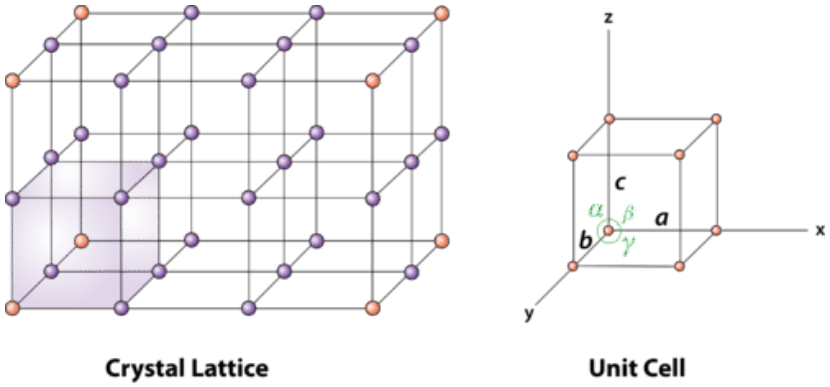


Figure 1: A figure showing a crystal lattice (left) consisting of multiple unit cells (right). Here, a, b and c indicate the lengths of the edges of the unit cell, and α, β and γ the angles between said edges. From [7].

2 Background & Theory

This section provides the theoretical background needed to understand the methodology used in this thesis. The topics presented here focus primarily on the chemical aspects of this research and provide only the necessary chemical basis needed for the computational approach discussed in the rest of this thesis. This section first explores basic concepts of materials chemistry, particularly focussing on the unique tunable properties of 2D perovskites. Next, the section provides an overview on how to run Molecular Dynamics simulations, which is necessary in understanding the computational analysis of the 2D perovskites. Finally, this section explains machine learning models relevant to this research, which will later be used for the analysis of the 2D perovskites.

2.1 Materials

This section explains the 2D perovskites and their properties, and elaborates on why they are the main focus in this research. First, the section introduces crystal structures and perovskites, after which this section discusses the 2D perovskites and their unique properties, and finally highlights the applications of said properties in, for example, producing energy.

2.1.1 Crystal Structures

Crystals in chemistry are defined as solids in which the atoms are arranged in such a way that it creates a periodic pattern that can be repeated infinitely in three dimensions [8, 9]. The smallest repeatable unit of said atoms of the material is called a unit cell. The unit cell can be placed in a crystal lattice for it to be repeated, thus creating the material; this periodic property of crystals makes it possible to describe an entire crystal based on only a single unit cell. An example of both a unit cell and crystal lattice can be seen in Figure 1. Note that this figure represents the unit cell, and thus the lattice, as cubic, meaning all edges are the same length, and all angles between the edges are 90°. However, this is usually only the case in an ideal situation; in reality the edges are not of equal length nor are the angles all 90°.

The ordered structure of the crystals makes the chemical properties of materials based on such crystals predictable. Since crystals are predictable and commonly found in nature, crystal-based materials are widely researched in both experimental and computational fields to better understand the behaviour of such materials, and be able to use such knowledge for real-world applications and technological advancements [10].

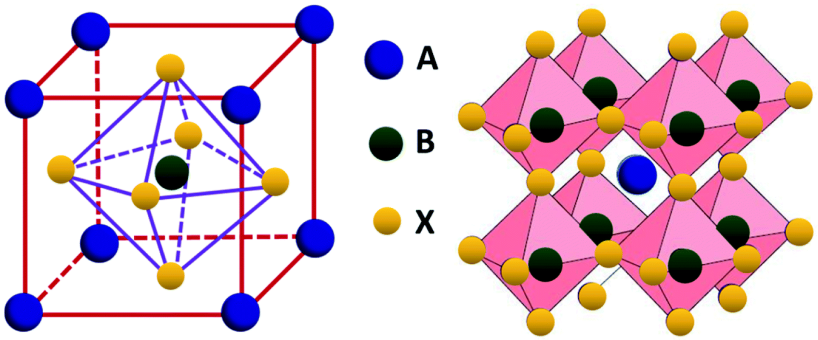


Figure 2: A visual example of a unit cell of a perovskite material (on the left). The blue (A), black (B) and yellow (X) spheres indicate different ions, where A represents a large cation, B a smaller cation and X an anion. The extension of this unit cell in a crystal lattice is shown on the right. From [13].

Crystallographic Information Files (CIFs)

Information of the unit cell and crystal lattice of a material is digitally stored in Crystallographic Information Files (CIFs) [11]. Information that can be found in CIFs includes the number of atoms in a unit cell, the element of each atom, the position of each atom in the unit cell, as well as the lengths of the edges and angles of the unit cell and the density; density in this context means the mass of the unit cell divided by its volume. The edges of the unit cell are measured in angstrom (Å), which is 10^{-10} metres. CIFs also contain information describing how atoms are repeated and positioned in the unit cell, which is necessary to accurately reproduce the crystal based on the unit cell; in the field of chemistry this is referred to as symmetry.

As mentioned above, the material is a crystal, thus making it periodic. The periodicity means the unit cell is repeated in the crystal lattice an infinite amount of times. In crystallography, atomic positions within a unit cell are described using fractional coordinates, which are defined with respect to the edges of the unit cell [8]. Any repetition of the unit cell can be represented by a difference of integers of the fractional coordinates along the unit cell edges. This is useful because it makes it easier to compare between different unit cells of different materials.

2.1.2 Perovskites

Perovskites are a class of materials which can be recognised by their unique repeatable pattern creating the crystal structure [8]. For perovskites, the formula of this repeatable pattern can be written as ABX_3, where A, B and X are atoms or molecules, and X appears three times more relative to A and B. An example of this can be seen in Figure 2. Note that this figure might not seem to respect the ABX_3 pattern, but it does considering atoms are shared with the surrounding unit cells (not visible in the figure). Corner A-atoms contribute 1/8 each, X-atoms on the faces contribute 1/2 each, and the B atom lies entirely within the unit cell, resulting in the correct composition of ABX_3 [12].

Ions are species, like atoms or molecules, with a difference in charge. Species that lose electrons are called cations, which are positively charged, while the ones that gain electrons are called anions, which are negatively charged. In Figure 2, A is a large cation, B a small cation, and X an anion. A balance between the size and charges of these ions allow the structure to remain stable [14]. It is important to keep in mind, however, that the unit cell displayed in Figure 2 represents an idealised, perfectly cubic perovskite structure. In reality, the composition of a perovskite material is dependent on the choice of the A, B and X ions determines the resulting crystal structure, which can deviate from the ideal cubic structure.

The stability and distortion of the structure is dependent on the size of the ions in the unit cell. If, for example, the A-ions are small relative to the B- and X-ions, the cavity in which the A-ion sits will be too big, and thus the B- and X-ions will compensate and move closer to the cavity, thereby

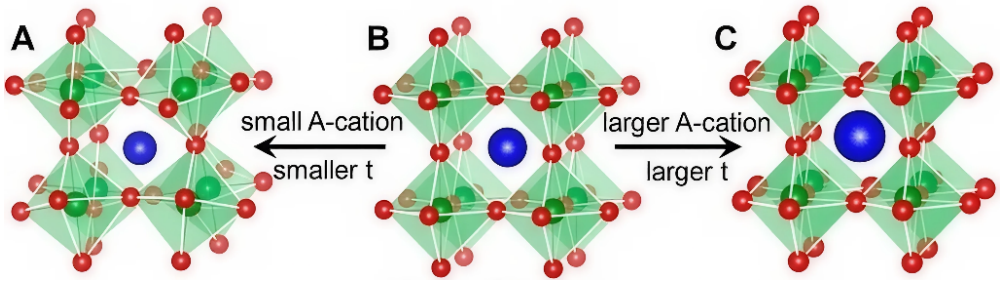


Figure 3: A visual example of different distortions in a unit cell depending on the size of the A-ion (the blue sphere). The cell in the middle (cell B) is the ideal structure, while the cell on the left (cell A) shows the cell tilting inwards due to ion A being small, and the cell on the right (cell C) shows the cell tilting outwards due to ion A being large. The t here describes the Goldschmidt tolerance factor. Upscaled; from [12].

tilting the unit cell inwards. On the other hand, if A is larger, the B- and X-ions will be pushed away from the position of the A-ions, and the structure will tilt outwards. An example of this is shown in Figure 3.

The extent to which the size of the A-, B- and X-ions cause distortions can be predicted by looking at the Goldschmidt tolerance factor (t) [12]. The formula for calculating the Goldschmidt tolerance factor is $t = \frac{r_A + r_X}{\sqrt{2}(r_B + r_X)}$. In this formula, t indicates the Goldschmidt tolerance factor, while r_A, r_B and r_X indicate the radii of the A-, B- and X-ions respectively. This formula shows that, if the radius, and thus the size of A, decreases, t will become smaller, meaning the structure will be tilted inwards, while a larger A leads to an increase in t, resulting in the structure tilting outwards. A value of t in-between 0.8 and 1.0 usually indicates a stable perovskite structure [12].

2.1.3 2D Perovskites

When the value of the Goldschmidt tolerance factor t exceeds 1.0, the structure will be tilted outwards and become unstable. This can result in the A-ions becoming too large and therefore being unable to fit in the ideal 3D perovskite crystal lattice. Because this ideal lattice is not able to be formed with the oversized A-ions, the structure stabilises by forming separate layers of BX-slabs, with the large A-ions in between these layers. The A-ions are now called spacer (cat)ions. When this happens, the original perovskite structure, which extended in three dimensions, now becomes slab-like, thus only extending in only two dimensions; these structures are now called *2D perovskites*.

2D Perovskite Formula

Rather than the ABX_3 formula for 3D perovskites, the general formula representing the crystal lattice for 2D perovskites is $A'_m A_{n-1} B_n X_{3n+1}$ $(m = 1, 2; n = 1, 2, 3, 4 \ldots)$, where A' indicates the spacer cations separating the slabs, A the cations inside the BX-layers in the slabs, B and X ions the framework of the separate layers, n the amount of BX-layers before being separated by spacer cations, and m the amount of spacer cations in between the BX-layers [3, 4, 12]. In the case of true 2D perovskites, where a single BX-layer is always separated by spacer cations, $n = 1$, which simplifies the above formula to $A'_m BX_4$ $(m = 1, 2)$. An example highlighting the difference between a regular (3D) perovskite, and a true 2D perovskite, meaning $n = 1$, can be seen in Figure 4. Note that the spacer ions as seen in Figure 4, are, in the case of 2D perovskites, usually molecules rather than atoms. This is because, even though the calculated Goldschmidt tolerance factor t for some large A-atoms might exceed 1.0, this does not guarantee a 2D perovskite being formed, and even if it does, atoms are usually not sufficiently large to stabilise the 2D structure; hence the typical usage of molecules as A-ions [16].

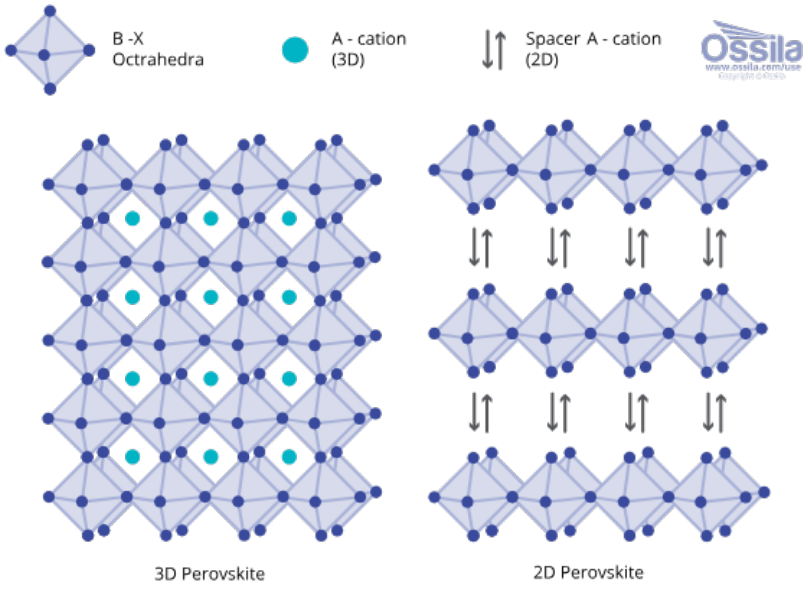


Figure 4: A visual example showing the difference between the ideal structure of a 3D perovskite (on the left) and the regular structure of a 2D perovskite (on the right). Adjusted from [15].

Inorganic vs. Organic

In the 2D perovskite unit cells, the BX-layers are inorganic, meaning they are not made out of carbon-hydrogen groups, but rather, for example, metal atoms, allowing for stronger bonds between the atoms within the inorganic layers, leading to strong, stable and continuous networks. Because of these strong networks, the electrons within this network can easily be shared across atoms by moving through the lattice. Therefore, electrons within the inorganic layers can move freely, therefore giving the inorganic layers (semi)-conducting properties. More information regarding semi-conduction properties are given in Section 2.1.4.

On the other hand, the spacer A-ions are organic, which means that they are made out of carbon-hydrogen groups and usually exist in molecules, kept together by relatively weak forces making them more flexible, rather than exist in strong layers or networks as is with inorganic components [16]. The lack of strong and stable networks means that the electrons are bound to their place within the molecules. Therefore, with the spacer ions consisting of organic molecules, the electrons within the spacer molecules are unable to move freely, thus giving the spacer molecules insulating properties [3, 17]. More information regarding insulating properties are given in Section 2.1.4. Note that, since there is a large variety of organic spacer molecules, many different 2D perovskite structures are possible.

2.1.4 Electronic Structure and Optoelectric Properties of 2D Perovskites

This section introduces important electronic and optoelectric concepts regarding 2D perovskites. The section explains the importance of the band gap and how the layered 2D perovskite structure gives it unique and tunable optoelectric properties.

Band gap

In a material, there are varying ranges of energy levels, also called energy bands, which electrons can occupy. These energy band ranges can be split up in two: low energy bands, which are usually fully occupied and where the electrons usually sit, and the higher energy bands, which are (usually) not filled [18]. In the case in which they are partially filled, the electrons can move freely; this is what

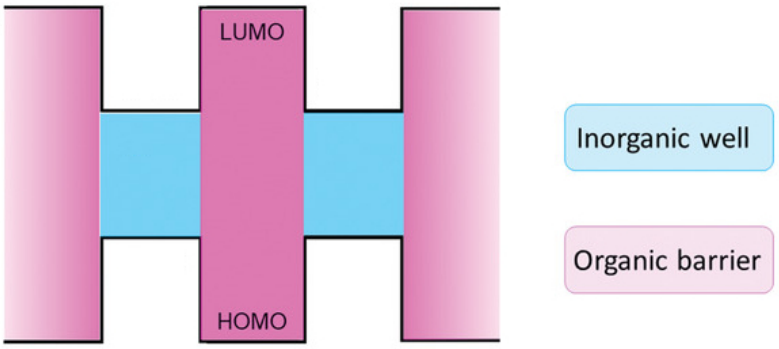


Figure 5: A visual example showing multiple quantum wells caused by a layered structure. The LUMO refers to the lowest unfilled energy band, while the HOMO refers to the highest filled energy band. Adjusted from [23].

allows a material to conduct electricity: the free movement of electrons in the upper ranges can carry an electric current through the material [19].

The gap in between the highest filled energy band and the lowest unfilled energy band is called the band gap [20]. The size of the band gap influences how well a material can conduct electricity: if the band gap is very small, electrons can easily jump between the highest filled and lowest unfilled energy bands, but if the band gap is large, electrons are stuck within the highest filled energy band, unless they are given a relatively large amount of energy to jump the gap; this is usually done by photons [21]. Materials with a large band gap are called insulators, and materials with a small band gap are called semi-conductors.

When a photon with an energy of at least the band gap is absorbed by a material, an electron can be excited from the highest filled energy band to the lowest unfilled energy band. If an electron is excited from one band to another, it leaves behind a hole. The excited electron and the left-behind hole travel through the material, thereby being able to produce an electric current. This demonstrates the process used by solar cells, and therefore highlights the importance of the size of the band gap in determining how effectively a material can convert sunlight into electrical energy [19].

Optoelectric Properties

A quantum well is a structure in which electrons, as well as the holes the electrons can occupy, are confined by surrounding energy barriers [22]. As mentioned in Section 2.1.3, in 2D perovskites, usually the BX-layers are inorganic, and the spacer ions are organic. The combination of the two creates a layered structure that behaves like a quantum well. In such a quantum well, electrons, as well as the holes they can occupy, are confined to the inorganic BX-layers, while the organic spacer ions act as barriers between those layers, thereby restricting the movement of the electrons and holes. Because of this confinement, the distance between the conduction and valence band, and thus the band gap, is dependent on the thickness of the inorganic layers, as well as the size and type of the organic spacer ions [12]. An example of such a layered structure behaving as quantum well can be seen in Figure 5.

As mentioned in Section 2.1.3, there is a large variety of spacer molecules that can be used to create 2D perovskites. Therefore, the organic spacer molecule barriers can easily be changed, depending on what kind of electronic and structural properties are desired. This is what makes 2D perovskites especially interesting: their properties can be slightly adjusted, by changing the organic spacer molecules. Changing the thickness can be achieved by changing the type of atoms in the BX-layer, although the types of inorganic atoms which allow for a stable 2D perovskite material are limited [24].

2.2 Molecular Dynamics Simulations

Chemical materials can be computationally researched using Molecular Dynamics (MD) simulations. MD simulations model the atoms in the material, and their interactions between one another, based on mathematical models.

This section first elaborates on the requirements needed to perform MD simulations, after which it explains the mathematical models used to model the interactions between atoms. Finally this section expands on the thermodynamic components required during the simulations.

2.2.1 Components of Molecular Dynamics Simulations

To be able to perform Molecular Dynamics simulations, several components are required.

First, MD simulations require a computational representation of the unit cell of the material. This computational representation should contain all the atoms and their positions, the bonds, angles and dihedrals (which describe rotations around bonds) of the atoms, as well as the sizes and angles of the unit cell. The atoms, bonds, angles and dihedrals are split up in groups, or types, to be able to differentiate between chemically different interactions within the material. For example, a carbon atom in a linear chain can have different electrical properties than a carbon atom in a ring-structure.

Second, the mathematical models required to describe the interactions between atoms in MD simulations are called force fields. Specifically, these force fields model the interactions between bonded atoms, with the use of mathematical formulas describing the forces between pairs of atoms. Alongside these bonded interactions, non-bonded interactions need to separately be defined, which describe long-range effects between atoms that are not directly bonded.

Lastly, the MD simulations require a simulation engine that uses the defined force field models and simulation settings to determine the positions of the atoms at each time step in the simulation, and thus how the system is supposed to evolve over time. This requires defining the amount of time steps the simulations should take, the total length of the simulation, as well as how the temperature and pressure are controlled during the simulation. These components together define how the MD simulations, with the use of the simulation engine, are supposed to function.

2.2.2 From Unit Cell to Simulation Engine

Performing MD simulations requires a computational representation of the unit cell. However, a single unit cell by itself is not a chemically accurate representation of a material. The simulation engine used in this research is LAMMPS [5]. It should be noted that LAMMPS requires the coordinates of the atoms to be in Cartesian coordinates, while the positions of the atoms as usually defined in the CIFs are in fractional form.

To construct a realistic simulation box, the simulation engine replicates the unit cell. Replicating the unit cell increases the number of atoms in the simulation box, which results in an overall larger system that more closely resembles the crystal lattice of the material; these replications can be done in all three dimensions. Another way of creating a more realistic simulation box is by using periodic boundary conditions (PBC). PBC describe how the atoms at the boundaries of the box behave. If PBC are turned off atoms at the boundaries of the box only interact with atoms inside of the box, thereby creating a finite system. While, if PBC are turned on, the box is treated as being infinite, thus allowing the atoms at the boundaries to interact with atoms from neighbouring replicas of the box.

2.2.3 Force Fields

The simulation engine by itself is not able to model the interactions between atoms; for that, force fields are needed. Force fields are a collection of mathematical equations describing the interactions between atoms by defining how atoms influence one another when they are brought closer or moved

further away in the system. These equations define the potential energy of the system, which then allows the simulation engine to determine how the atoms in the system evolve over time.

Different Force Field Models

There are different types of force fields models, which all have the same goal, namely modelling the interactions between atoms. However, between each of the force field models, the definitions of the mathematical formulas and parameters is different.

Ideally, a force field specific to each unique material is created, since each unique material can have material-specific chemical interactions between the atoms. However, due to the complexity of creating force fields, most force fields used for MD simulations are general force fields. General force fields are force fields created to describe general chemical interactions, while allowing for material-specific behaviour through the introduction of parameter values specific to the material being simulated. These general force fields can therefore be optimised for a specific system based on the parameters that are used.

Well-known general force fields include CHARMM and AMBER [25–28]. These force fields have been constructed based on biomolecular systems, like systems with proteins, although they can also be used for simulating non-biomolecular systems. When used for non-biomolecular systems, the general force fields are typically used to analyse different behaviours between systems, rather than simulating highly accurate structures and extract exact material-property values.

Non-bonded Interactions

The force fields describe the interactions between bonded atoms. However, the effects of interactions between non-bonded atoms need to be taken into account as well. To account for these interactions in MD simulations, Buckingham or Lennard-Jones potentials are typically used. The Buckingham and Lennard-Jones potentials are mathematical descriptions of how pairs of atoms attract and repel each other, based on the distance between the two atoms. Both Buckingham and Lennard-Jones potentials describe non-bonded interactions, although the way these interactions are represented differs. Buckingham potentials are used to describe short-range non-bonded interactions, while the Lennard-Jones potentials are used for long-range non-bonded interactions. Alongside the short- and long-range interactions, force fields usually also include additional interactions between atoms based on their atomic charges [29].

2.2.4 Ensembles

Using MD simulations, changes in the material under varying conditions, like changing the temperature or pressure, can be examined. These settings are defined through the use of ensembles. Ensembles describe which properties of the system are kept constant during the simulation, while the behaviour of the other properties can be observed.

Two commonly used ensembles are NVT and NPT ensembles. In NVT ensembles, the number of atoms (N), the volume (V) and the temperature (T) is kept constant. This is typically used to examine behaviour of the material at varying pressures, without changing the size of the box. On the other hand, NPT ensembles keep the number of atoms (N), the pressure (P) and the temperature (T) fixed during simulations, while allowing the volume of the box to change [5]. Therefore, NPT ensembles are typically used to examine structural changes of materials at different temperature conditions; the NPT ensembles are also used in this research.

2.3 Machine Learning Models in Material Science

Supervised learning methods are used in material science to, for example, be able to find new material compositions, predict properties of materials, and to explore relationships between structural features and physical properties of materials [30].

Many material properties, like structural, thermal and energetic properties are represented by continuous values rather than discrete values. To predict continuous values, regression models can be used to capture the relationship between the structural features and physical properties of a material. This allows for the prediction of material properties for both known and new materials, without requiring experimentally determining such properties, which can be difficult and time-consuming due to high costs and needing specialised equipment. Regression models try to learn the relationship between features and targets. There are different types of regression models; some of these models assume the relationship between the features and the target is linear, while other models attempt to capture non-linear feature-target relationships. This thesis considers three linear regression models, as well as one another model, for predicting properties of 2D perovskites.

Linear Regression

Linear regression models make the assumption that the predicted target can be expressed as a linear combination of the provided input features. The goal is to determine the weights of the features, or coefficients, such that said weights lead to optimal target predictions, based on the provided data [31]. The following three sections explain the linear regression methods in this research, and how each of these methods approaches the determination of the coefficients.

Ordinary Least Squares

Ordinary Least Squares (OLS) is a form of linear regression which estimates the weights of the features, the coefficients, by minimising the sum of the squared differences between the values of the predicted and true target values [31].

In this approach, the weight of each feature is influenced by how that specific feature contributes to reducing the prediction error of the model as compared to the other features. However, this means that when the number of features approach the number of available target data points, the features are unable to provide the model with enough independent and unique information to consistently reduce the prediction error [32]. This makes the predicted coefficients highly sensitive to noise, and thus makes the model unstable and more likely to overfit. Therefore, when using many features, OLS is often not suitable to learn the feature-target relationship.

LASSO

Another form of linear regression is LASSO (Least Absolute Shrinkage and Selection Operator). LASSO introduces an L_1 regularisation term to help find the optimal coefficients. The L_1 regularisation term penalises large coefficients, and shrinks coefficients which contribute little to the predictions to zero [32]. Feature selection is already performed by using the L_1 regularisation term since shrinking coefficients to zero essentially removes the features from the model [31]. In contrast to OLS, which can become unstable when many features are used, LASSO performs well in cases where many, potentially redundant or strongly correlated, features are used.

Ridge

Ridge is also a form of linear regression using a regularisation term. Rather than LASSO using an L_1 regularisation term, which shrinks coefficients that contribute little to the prediction *to* zero, Ridge uses an L_2 regularisation term, which shrinks large coefficient values *towards* zero. Note that, L_2 does not force coefficients to zero, as opposed to the L_1 regularisation term LASSO uses [31, 32]. This makes Ridge also suitable in cases where potentially redundant or strongly correlated features are used, as well as in cases where it is desirable to use all features in the model.

Random Forest

Random Forest is a non-linear regression method based on an ensemble of decision trees. Instead of using a single model to learn the feature-target relationship, Random Forest uses the predictions of many different decision trees that are trained on different subsets of the dataset [33]. This allows

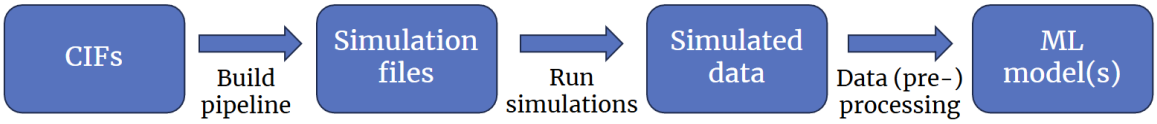


Figure 6: A visual overview of the workflow of this research.

Random Forest models to capture non-linear and more complex relationships between features and the target. This is also the reason the Random Forest method is usually less sensitive to noise and overfitting. Therefore, Random Forest is suitable in cases where the data is noisy or the feature-target relationship is expected to be more complex or non-linear.

3 Creating & Running Simulations

The goal of this research is to create simple ML models that try and predict physical properties of 2D perovskites. This requires simulated data from different 2D perovskite materials. However, datasets containing the simulated material properties of 2D perovskites are not easily available. Existing studies often focus only on a subset of many 2D perovskite materials [34, 35]. The simulation approaches between studies also vary, so it is not reliable to combine the information regarding simulated 2D perovskite materials from different papers. Besides, there currently is no widely available technique that allows for the automatic generation of the files needed to run the simulations for different 2D perovskites. However, while simulated data is not widely available for 2D perovskites, the Crystallographic Information Files (CIFs) are.

As described in Section 2.1.1, CIFs describe the type of element and positions of the atoms, as well as some information about the unit cell itself, like the lengths of the edges. Although these CIFs are widely available for 2D perovskites, essential information (see Section 2.2.3) required for running MD simulations, like the typing of atoms and parameterisation, are missing. The CIFs will therefore be used to extract structural information, to then be able to convert this structural information to simulation-ready files; this conversion can be done through the use of an automated pipeline, which we create during this research project. All CIFs used during this research are gathered from the Cambridge Crystallographic Data Centre (CCDC) [36].

This section describes the creation of the automated pipeline needed to convert CIFs to simulation-ready files. The section after presents the processing and analyses of the data gathered from the results of these simulations. An overview of the workflow of this research can be seen in Figure 6.

3.1 Extraction of Structural Information from CIFs

Seeing as the unit cell in the CIFs contain both BX-layers and organic spacer A-molecules, these components first need to be identified, and separated, before being able to perform force field parameterisation. This information can then be used to reconstruct the unit cell in a format that is suitable for the LAMMPS simulation engine [5]. Since this process is not yet automated, we create a pipeline, which automatically converts CIFs to simulation-ready files.

Identifying the Atoms

First, the organic spacer molecules, as well as the atoms in the inorganic layers, are identified and extracted. Seeing as the inorganic atoms are not bonded, but rather exist in an ionic lattice, the extraction of the inorganic atoms, and their positions, can be done by looking up the inorganic atoms in the CIF, and copying their information. The organic spacer molecules are more difficult since a single unit cell can contain several of the same spacer molecules with different symmetry configurations (see Section 2.1.1). In practice, this means that the same molecule can appear multiple

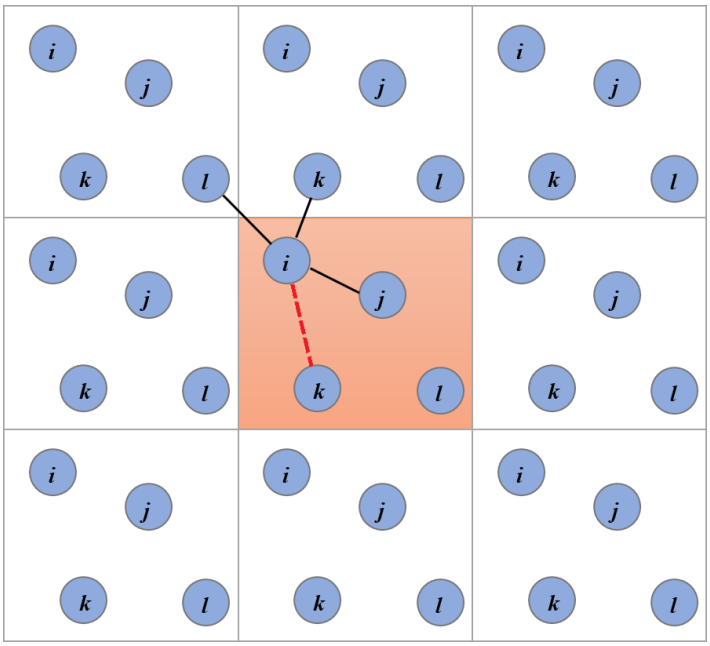


Figure 7: A visual representation of periodicity similar to a crystal lattice, where each of the nodes can resemble atoms in a molecule, and each of the edges bonds between atoms the bonds between the atoms of the molecule. The dotted red edge highlights the difference in distance between periodic images. Adjusted from [37].

times in different orientations, like flipped or rotated, within the same unit cell. This is an issue because each individual spacer molecule needs to be extracted to perform force field parameterisation.

The solution is to construct a heuristic-based graph to determine which atoms belong to which molecule. The heuristic is based on a cutoff, meaning that two atoms are considered bonded if the distance between said atoms is below a certain threshold; the cutoff threshold is calculated as the sum of the bond radii of the two atoms, with an additional tolerance factor of 0.4. This approach is able to correctly group atoms into their respective spacer molecules and therefore correctly identifies each molecule.

However, the heuristic is not able to capture the connectivity within each spacer molecule; the edges within the graph—the bonds between the atoms in the molecules—are therefore incorrect. Adjusting the heuristic did not fix this seeing as the distances of the atoms within molecules can vary between all the unique materials. Therefore, each unique spacer molecule within a material can be determined using this approach, but identifying the connectivity of the atoms using the atomic radii and average bond lengths is unreliable. Note that this two-step approach is used for clarity; in the pipeline both steps can be combined into a single step.

Ensuring Complete Molecules within a Unit Cell

After identifying the atoms belonging to each spacer molecule, each of the atoms need to be parametrised for the force field, which requires the connectivity between these atoms to be inferred. OpenBabel can easily infer this connectivity by using the coordinates of the atoms [38]. However, because the CIFs describe periodic crystal structures, it is not guaranteed that all atoms of a single spacer molecule are located within the same unit cell. Instead, parts of the molecule might be wrapped around the edges of the cell. A visual example of this can be seen in Figure 7. Figure 7 shows a periodic lattice. This lattice can be seen as a crystal lattice, where each of the 9 boxes are replicated unit cells; the CIF contains only one unit cell, the orange cell in the middle for example.

This lattice contains nodes i, j, k and l, where atoms j, k and l are all connected to node i, but not any other nodes. This figure demonstrates the periodicity issue: not all of the connected nodes are within the same cell. In particular, looking at node i in the orange unit cell shows some of its connections crossing over into the surrounding unit cells. While the true distance between, for example, nodes i and k is given by the black edge, the CIF only contains the orange cell, and a heuristic-based approach will only see the longer distance for the node within the orange unit cell, as indicated by the dotted red line. Therefore, before being able to infer connectivity, the nodes in the cell are all moved based on the periodic image in which they are closest to a chosen reference node, thus essentially shifting an entire connected graph to lie within a single unit cell. This reference node can be any node in the cell, since the distance from a random node to any of the other nodes of the correct graph will always be shorter than the distance needed for traversing a periodic image to a node belonging to a different graph.

Therefore, all unique spacer molecules can be identified through a heuristic graph-based approach, while determining the connectivity of the atoms within each molecule requires accounting for the periodicity of the crystal lattice of the materials, which is done by shifting all atoms a certain distance based on a reference atom.

3.2 Force Field Parametrisation

The next step is to parametrise the identified spacer molecules using the AMBER general force field. This force field is chosen because, as described in Section 2.2.3, a general force field is needed for handling many different materials without manually tailoring a specific force field to each. Besides, AMBER comes packaged with additional tools useful for parametrising the atoms [26–28].

First, OpenBabel is used to infer connectivity between the atoms [38], while antechamber and parmchk2—belonging to the AMBER general force field—are used to parametrise the atoms of the spacer molecules to reassemble all the spacer molecules and inorganic atoms into a single cell, including the lattice parameters like the lengths of the edges of the unit cell. The LAMMPS simulation engine can then use the cell to perform the simulations.

Unit cells can contain spacer molecules with different symmetry configurations, but antechamber and parmchk2 expect a single molecule. To solve this, a single spacer molecule is first parametrised, after which its parametrised values are also used for the remaining symmetry configurations, which is possible since each symmetry configuration represents the same molecular structure and therefore share identical force field parameters. It is important to note that the automatic determination of the charges of the spacer molecules can not be automated; these charges are therefore manually provided.

3.3 Non-Bonded Interaction

The last step requires the definition of the Buckingham and Lennard-Jones potentials to model the interactions between non-bonded atoms, as described in Section 2.2.3. In the case of 2D perovskites—which are the focus of this research paper—the organic atoms within a single molecule will be described using the short-range Buckingham potentials, while interactions between the atoms in the inorganic BX-layer with themselves, as well as with atoms from the organic spacer molecules, are described using Lennard-Jones potentials.

The Buckingham potentials are automatically generated by tLeap alongside the LAMMPS-compatible unit cell; this means the short-range organic-organic interactions are automatically defined. This leaves the long-range organic-inorganic, and inorganic-inorganic interactions to be described using the Lennard-Jones potentials. The values for these Lennard-Jones potentials can only be accurately be determined experimentally. Although, not much previous work is available regarding the use of Lennard-Jones potentials for inorganic-organic systems.

However, there are some papers, like [34], that experimentally determined the Lennard-Jones potentials for some inorganic-organic systems. But, this means that all atoms in the system are now tied to the type of elements used in said paper, since no parameter values are available for any other type of elements. For example, the paper does not include Lennard-Jones potentials for bromine (Br) atoms, meaning all CIFs containing bromine can not be used; the only inorganic atoms in said paper are lead (Pb) and iodine (I). A different paper [35] does report Lennard-Jones potentials for Br and Pb atoms. However, since the reported Pb parameters differed between the two papers, only the values from Fridriksson et al. are used to avoid mixing parameters from different sources. Besides that, a lot of values in said paper are the same across atom types; for example, a carbon atom in a simple chain and a carbon atom in an aromatic system had the same value for Lennard-Jones potentials in the paper. Using the same Lennard-Jones potential values for different atom types will likely not affect the comparison between materials as is done during this research, although it does introduce the possibility of the simulated material being less chemically accurate.

Nonetheless, due to long-range non-bonded interactions being crucial for the accurate simulation of organic-inorganic systems, like the 2D perovskites, the values for the Lennard-Jones potentials from [34] are used. The Lennard-Jones values, together with the already generated Buckingham values, model the non-bonded interactions of the atoms.

3.4 Conversion to LAMMPS

The parametrised spacer molecules, together with the inorganic atoms and lattice parameters, are used by tLeap, also belonging to AMBER, for the creation of a single unit cell, appropriate for LAMMPS simulations. It is important to note that for the parameterisation in antechamber, the overall charge of the spacer molecules is needed.

After the reassembly of the unit cell containing all spacer molecules, inorganic atoms and lattice parameters, the cell needs to be converted to a format that can be used by LAMMPS. Since converting the files from an AMBER-compatible format to a LAMMPS-compatible format is not possible, an intermediary step is added to first convert AMBER files to InterMol-compatible files, before using InterMol to write everything to a LAMMPS-compatible file [39]. After the conversion, a simulation-ready file containing the reassembled unit cell is generated, which can then be used by the LAMMPS simulation engine to perform simulations.

3.5 Running Molecular Dynamics Simulations

The pipeline as described in the previous sections is able to convert structural information from CIFs to simulation-ready files. Before the MD simulations can be performed the LAMMPS simulation engine requires the definition of several settings in order to model how the system changes over time. These settings include thermodynamic settings, like the chosen ensemble, the duration of the simulation and the size and periodic boundary conditions of the system. This section highlights which LAMMPS settings are used for simulating the 2D perovskite materials.

Thermodynamic Settings

Since one of the goals of the simulation is to examine structural changes caused by varying the temperature, the chosen ensemble for simulating the 2D perovskites is the NPT ensemble, which keeps the pressure constant, rather than the volume. This way, the change of volume of the box can be examined under varying temperatures.

The temperatures used in this research range from 200K to 350K, in steps of 10K. This means, in total 16 NPT simulations are performed per unique material. The temperature range converted to degrees Celsius would be about -73 to 77 degrees Celsius. Realistically, 2D perovskites would not endure such extreme temperature, but using such a wide range can allow for temperature-related effects to be better observable. Seeing as the goal of this research is to identify the behaviour of the

materials relative to each other, rather than obtaining highly specific structural properties, this is a suitable approach for this research.

Simulation Boxes & Period Boundary Conditions

For the simulation of 2D perovskite materials, orthogonal simulation boxes with 3 x 3 x 3 replications of the unit cells are used. Usually, larger simulation boxes are preferred since they more accurately represent the crystal structures and better capture long-range interactions. Using a 3 x 3 x 3 simulation box surrounds the central unit cell in all directions, thereby improving the representation of the crystal structure. Due to the limited available time for this research, as well as the goal of this research being the identification of subsets of 2D perovskites with certain properties, rather than highly specific property information, the usage of a 3 x 3 x 3 simulation box is suitable. To further aim for a more representative crystal structure, periodic boundary conditions are used for the simulations, so that atoms at the boundaries of the simulation box interact with atoms in neighbouring replications of the system, rather than being cut off by the edges of the box.

Simulation Duration & Time Steps

The MD simulations evolve over time using fixed time steps. Typically, either one or two femtoseconds (10^{-15} seconds) are chosen for each time step, and seeing as larger time steps may cause instabilities during the simulations, a time step of one femtosecond is used.

Before the simulations are started, there is an initial equilibration phase where the system relaxes from its initial state to a more stable state. After this initial equilibration phase finished, the simulation using the first temperature, namely 200K, is started. The simulations are then performed sequentially for increasing temperatures, where the final configuration of each temperature run is then used as the starting configuration for the next run. For each temperature, a simulation of 40 000 time steps is performed. During each of the different temperature simulations, the first 4 000 time steps are used as an additional equilibration phase in which the system is provided with extra time to stabilise per temperature. Therefore, at each temperature, LAMMPS records the properties of the system every 1 000 timesteps for a total of 40 000 time steps, with the exclusion of the first 4 000 timesteps corresponding to the equilibration phase.

Recorded Properties

At each 1 000 time steps, in femtoseconds, of the MD simulations, several structurally and physically relevant properties of the simulated material are recorded by the LAMMPS simulation engine. LAMMPS outputs the total potential energy of the system, as well as the volume of the simulation box and the positions of all atoms in the system. Which of these properties are relevant for the machine learning models is discussed in the next section.

Restarting Simulations

In case of unexpected interruptions, the simulations are manually restarted. Under normal circumstances, the box of atoms is initialised and separately equilibrated, after which the equilibration, as described above, for the 200K run is performed. The following temperature runs then use the final positions of the 200K run as their starting positions. This works, seeing as structural changes of the material happen gradually with an increase in temperature, meaning that the previous temperature run provides a good starting position for the next run.

However, simulations are only allowed to run for up to 24 hours on the supercomputer used in this research, which means that if simulations take longer, they would be terminated before finishing all the temperature runs. These are then to be manually restarted to continue from the last temperature run at which they have stopped. In practice, a complete simulation of a single material over the entire 200K-350K temperature range usually takes between 10 and 20 hours. This means that the box of atoms might not yet be correctly relaxed at the start of the new temperature run. Despite this, since

there is an initial equilibration step, as well as the equilibration phase during the first 4 000 timesteps of each temperature run, this effect is expected to be negligible.

3.6 Assumptions

The previous section explains the approach that this research uses to perform MD simulations. However, throughout that section, several assumptions are introduced. This is difficult to avoid since representing materials or molecules with a physical meaning in a computational form requires partially abstracting away from this physical meaning to simplify the system and make it computationally suitable. The main assumptions relevant to running the MD simulations for the 2D perovskites in this research paper are summarised and justified in this section.

Charges

All atoms in a material have their own partial charges. These partial charges are influenced by the type of atoms in the material, the neighbouring environment, and its electronic interactions. Ideally, these charges are determined through quantum-mechanical calculations. However, in this research, the charges are approximated through a simplified approach using antechamber [26–28], which is deemed suitable since small differences in the partial charges will likely not influence the trends and behaviour of the overall material by much.

Force Field (Parameters)

A general force field is used to run the MD simulations, which requires force field parameters to be assigned. These parameters are also determined through antechamber, which tries to assign parameters based on the given molecular structure. However, the usage of a general force field and the assignment of force field parameters in itself will always be an approximation, since, ideally, a new force field would be created specialised for each unique structure, and each atom would get its own uniquely calculated force field parameters. However, the ideal approach is not commonly used as this is time consuming and not generalisable. Instead, the use of general force fields is standard practice in material chemistry. This will likely not influence the predictions of the simple ML models by much, but nonetheless it is important to keep in mind, especially since this would be relevant when chemically accurate and precise predictions or analyses are needed.

Orthogonal Boxes

The 2D perovskites are simulated using orthogonal boxes. This simplification is required since determining the two directions in which the 2D slab of the perovskite extends is difficult to be determined programmatically due to the lack of available information in the CIFs. Besides, using orthogonal boxes also assumes 90° angles between all the edges of the box, which is not always the case. About one quarter of the CIFs used in this research have at least one unit cell angle that deviates by more than 5° from 90°. Although these simplifications reduce the chemical accuracy of the structures, it is assumed this will not change the performance of the simple ML models by much, seeing as the predictions are based on relative trends between the materials, rather than precise geometrical features.

Simulation Duration

The simulations are performed over a time of 40 000 time steps, each with a duration of one femtosecond (10^{-15} seconds). In general, increasing the length of the simulations improves the statistical reliability of the results of the simulations, as it is more likely to average out noise. It is assumed 40 000 time steps are sufficient for this research because the initial part of each simulation is reserved for equilibration and not used for training the machine learning models. An increased amount of time steps could lead to more statistically accurate results, but it has the downside of requiring more computational power.

4 Simulated Data & Machine Learning Models

Before any models can be trained to predict properties of 2D perovskites, the first step is to identify which properties could possibly be predicted using machine learning models. This section therefore first discusses the different 2D perovskite properties extracted from the MD simulations. These properties are then analysed to identify which properties are suitable as ML targets, as well as to verify the target's contribution to the photoelectric effects in the system. The final section of the section describes the methodology for constructing and evaluating the ML models for predicting the targets.

4.1 Simulated Data

The simulations produce a range of physical and structural data. As described in Section 3.5, some simulations were too large to finish their simulations within 24 hours, and therefore had to be manually restarted; this happened for 16 unique materials. At each temperature, LAMMPS recorded the positions of the atoms every 1000 timesteps, with the exclusion of the first 4000 timesteps corresponding to the equilibration phase. Alongside the positions, LAMMPS also output total potential energy, the varying lengths of the box due to the changes in volume at different temperatures, the stress each plane of the box experiences, the short-range structure of the atoms, and the mobility of the atoms, all recorded at each timestep. However, not all of these properties gathered from the simulations are suitable as machine learning targets. A proper machine learning target, in this case, must be dependent on or influenced by the temperature, and must be linked to the structure of the 2D perovskites, since the CIFs contain structural information and the goal of this research is to create models that can differentiate between subsets of different 2D perovskite structures.

4.2 Target Selection

In an attempt to find behaviour suitable for machine learning, several properties gathered from the MD simulations are plotted. It should be noted, however, that for these plots, and the extraction of simulation data in general, all data up until the first 20000 timesteps per temperature run are not included in the calculation of the targets. This is done to ensure all systems have fully equilibrated, and no inconsistent or noisy data caused by incomplete relaxation of the system is used.

This section first examines whether there is a pattern to predict by looking at the variability of the potential targets across simulations to assess whether these calculated average values for each of the materials contain much noise, and finally it checks how the potential targets behave across different temperatures.

4.2.1 Average Potential-Energy Temperature Derivative ($C_p^{(\hat{U})}$)

The specific heat capacity at a constant pressure (C_p) describes the amount of heat required to raise the temperature of the material by one degree, at constant pressure, which describes how energy is distributed within the material [40]. The C_p is usually calculated using terms that introduce contributions that are not directly linked to bonding or structural information of the system. Since the goal of this thesis is to predict properties based on the static CIF features, using the standard definition of C_p is less informative. Instead, the average potential energy ($\hat{U}$) can be used since it is directly related to the bonding and structural configuration of the material and can immediately be extracted from the results of the MD simulations. Therefore, the first possible target $C_p^{(\hat{U})}$ is defined as the slope of the average potential energy $\hat{U}$ with respect to T at constant pressure. This target focuses on the potential-energy part and does therefore not represent the full specific heat capacity C_p.

This paragraph first examines whether there is a predictable pattern by looking at the variability of the average potential-energy temperature derivative across simulations, after which it considers

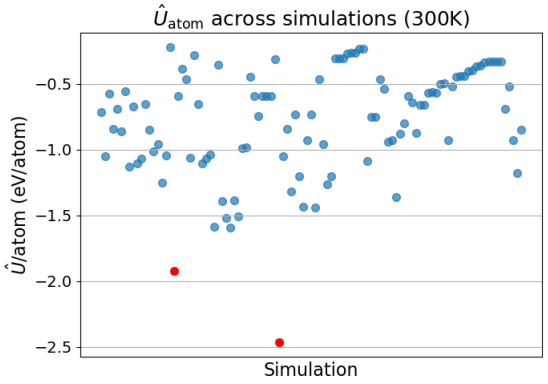


Figure 8: A scatter plot showing the average potential energy per atom at 300K for all simulations. For further analysis, the two red-coloured outliers are removed.

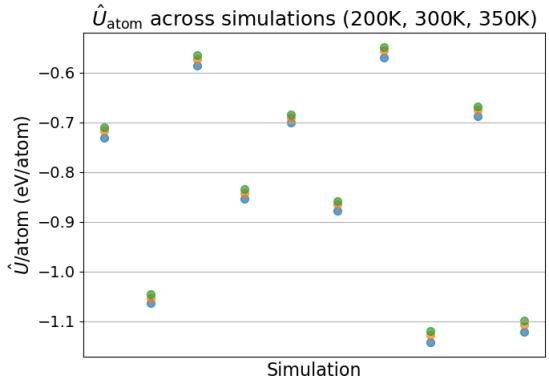


Figure 9: A scatter plot showing the average potential energy per atom at 200K, 300K and 350K for a subset of simulations.

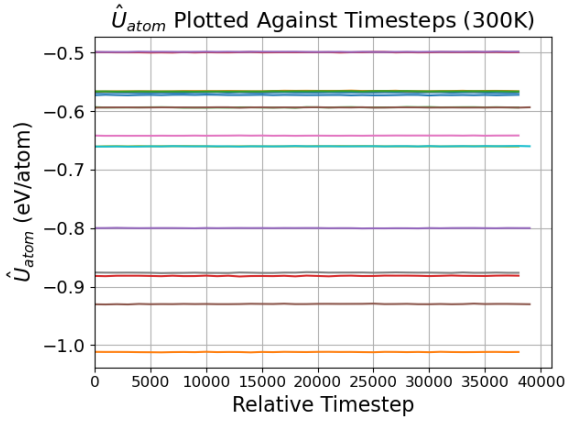


Figure 10: A graph showing the average potential energy per atom at 300K for a subset of materials, plotted as function of the relative time steps of the simulations. Each line represents a different simulated 2D perovskite material.

whether these average values are consistent, and finally it checks the slope's behaviour across different temperatures.

Average Potential Energy over Temperatures

First, the distribution of the average potential energy per atom values is examined. Figure 8 shows the values of the average potential-energy, at 300K, ranging from around -120 to around -20, indicating these average values vary across different materials. If the target is used, the red outliers in Figure 8 will be removed from the dataset.

Figure 9 shows the average potential energy per atom across three different temperatures, namely 200K, 300K and 350K. It is expected for the average potential energy to increase as the temperature increases, since higher temperatures cause increased atomic vibrations, meaning the system requires a higher amount of energy. This is indeed visible in Figure 9. Since the potential energy increases from lower to higher temperatures, the average potential-energy temperature derivatives across materials likely show similar behaviour, making them comparable and potentially suitable as machine learning

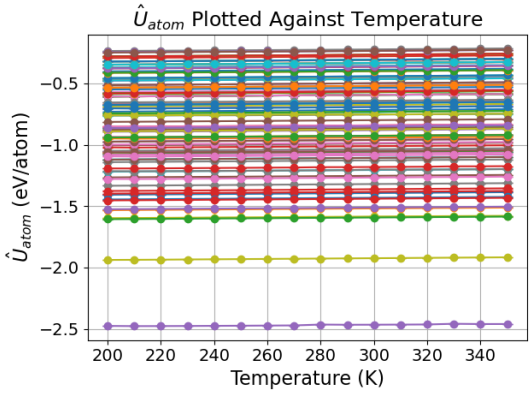


Figure 11: The average potential energy per atom (in eV/atom*) over the temperature (in* K*). Each line represents a different simulated 2D perovskite material. Here, two outliers have been removed.*

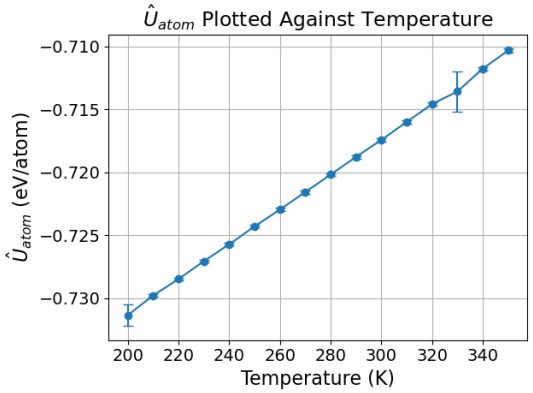


Figure 12: The average potential energy per atom (in eV/atom*) over the temperature (in* K*) for a single simulation.*

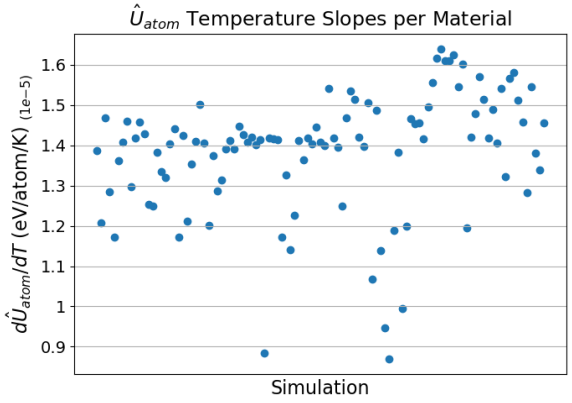


Figure 13: The single value for the average potential-energy temperature derivative per material. The two previously identified outliers have already been removed.

targets. Therefore, because the data falls within a similar range and order of magnitude—with the exception of two outliers—and there seems to be an increasing trend from lower to higher temperatures, this suggests the average potential-energy temperature derivative could be a suitable machine learning target.

Potential Energy over Time

The average values as seen in Figures 8 and 9 use averaged values. To confirm the stability of these average values, the averages are plotted against the timesteps of the MD simulations, at a single temperature; this can be seen in Figure 10. Figure 10 shows the averages of the potential energy per atom to remain stable over time. Therefore, the data points as shown in Figures 8 and 9 are more reliable.

Average Potential-Energy Derivatives as Target

Finally, the average potential-energy temperature derivative can be seen in Figures 11 and 12. The

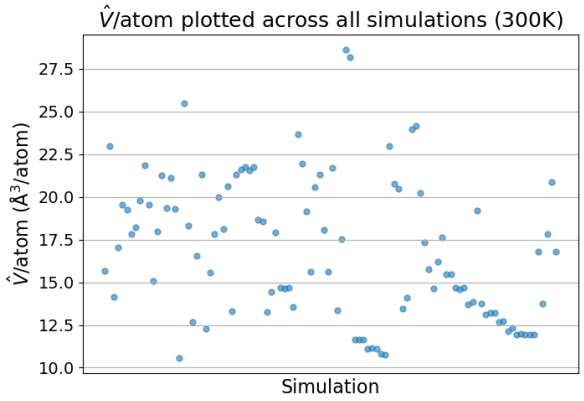


Figure 14: A scatter plot showing the average volume per atom at 300K for all simulations, except for the removal of a single outlier. The values are spread out, with a couple of values clustering together towards the bottom-right of the graph.

target is defined as the derivative of the average potential energy with respect to the temperature, and is determined by using OLS (see Section 2.3) to fit a line to the average potential energy per atom $\hat{U}_{atom}$ across the temperature range of 200-350K [41].

Figure 11 shows the average potential energy per atom plotted as function of the temperature for all materials, including two outliers. These outliers are the same as the outliers as visible in Figure 8, and are therefore removed. As expected from Figures 8 and 9, the slopes of the materials exhibit clear variability, while also showing an increasing trend with temperature. The increasing trend with temperature might not be clearly visible since the potential energies of the different structures vary over a range of temperatures. To clearly indicate this increasing trend, the average potential energy per atom over the temperature is shown in Figure 12. Looking at Figure 12 shows the slope for an individual material and confirms that the curves as seen in Figure 11 are indeed non-linear. The average potential-energy temperature derivative values per material can be seen in Figure 13 to confirm whether the values of the derivatives vary across simulations. Figure 13 shows variation between the values of the derivatives across different materials.

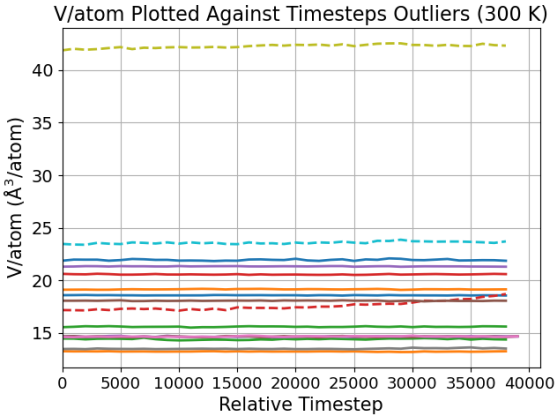
Therefore, since the average potential-energy temperature slopes show variability and a consistently increasing trend with temperature, the average potential-energy temperature derivative is a suitable machine learning target.

4.2.2 Average Volume per Atom ($\hat{V}_{\text{atom}}$)

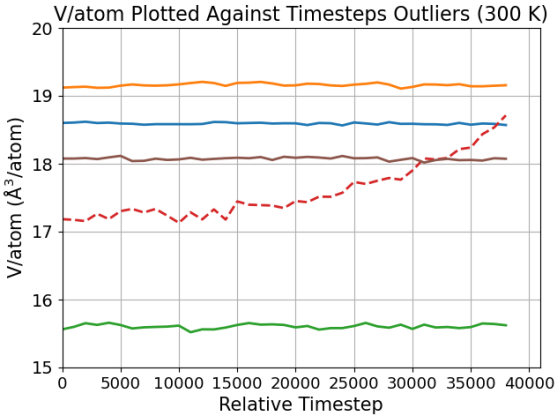
The second potential target is the average volume, per atom, ($\hat{V}_{\text{atom}}$) of the boxes. The average volume here is taken per atom, since the volume should be comparable between simulations. This section first examines whether there is a pattern to predict by looking at the variability of the average volume per atom across simulations, after which it considers whether these average values are consistent, and finally it checks how the average volume behaves across different temperatures.

Average Volume per Atom over Temperatures

To first see whether there is any variability between $\hat{V}_{\text{atom}}$ across simulations, $\hat{V}_{\text{atom}}$ is plotted against a single temperature. This can be seen in Figure 14. Looking at Figure 14 reveals there is a variability between the different values. The range and order of magnitude of the data points seem to also be in a similar range. Therefore, this could be an indication of the average volume per atom



(a) The regular lines indicate the average volume per atom, for a single temperature (300K), with a standard deviation below the threshold of 0.10. The dashed lines indicate some of the identified outliers exceeding the standard deviation threshold of 0.10.



(b) Same as Figure 15a, but then zoomed in on an area containing an outlier.

Figure 15: Graphs showing the accepted averages and identified outliers for the volume per atom at 300K plotted against the relative time steps of (a subset of) the simulations. Each line represents a different simulated 2D perovskite material. The outliers were filtered based on the standard deviations of the volume per atom over time exceeding 0.10.

being a target suitable for machine learning models.

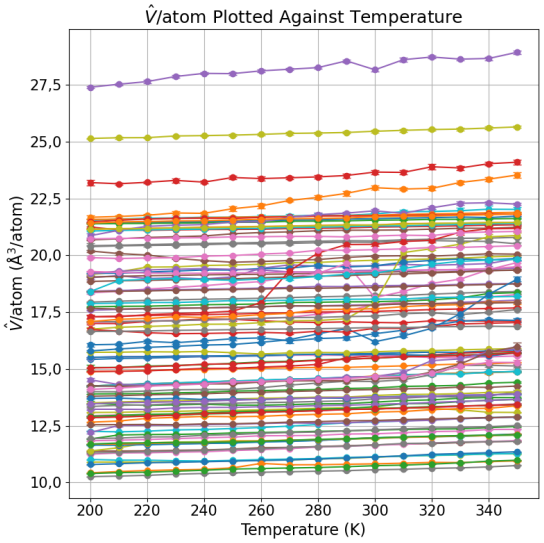
When comparing Figures 8 and 14 similar clustering of the data points is seen in the upper right and lower right corners of the Figures respectively. This is likely due to the numbering of the CIFs, originating from the CCDC [36], being ordered in a way that similar structures receive a similar number. Since the scatter plot goes through a sorted list of CIFs, it likely plots CIFs with similar numbers, and thus possibly similar structures, closer together.

Volume per Atom over Time

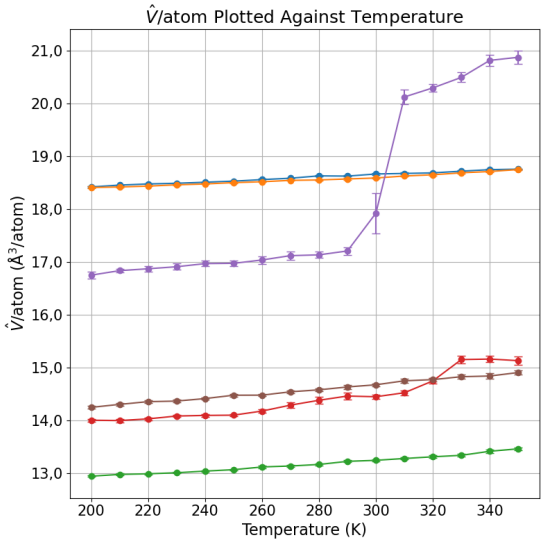
Next, it should be confirmed whether the average volume-per-atom values are stable so that they can be reliably used as a ML target. To confirm whether the values over time fluctuate, making the averages unreliable, the standard deviation of the volume per atom for all temperatures is determined, after which the mean of standard deviation is taken over all 16 temperatures to get a single mean standard deviation value per simulation. With a threshold of 0.10 for the standard deviation, six outliers are identified. The outliers for a single temperature, namely 300K, can be seen in Figure 15a. Since the values vary, the values over time in Figure 15a might seem consistent. However, looking at Figure 15b shows a zoomed-in version of Figure 15a, in which the instability of one of the average values is better visible. Note that the stability of the average determined by looking at the volume per atom over time, rather than over temperature, since a fluctuation in temperature does not have to be an indication of an outlier; a fluctuation could represent the physical behaviour of the box at said temperature.

Average Volume per Atom per Temperature as Target

Lastly, to ensure $\hat{V}_{\text{atom}}$ is a reliable target, $\hat{V}_{\text{atom}}$ is plotted against all the temperatures to show the behaviour of the potential target across different temperatures; this is shown in Figure 16a, which seem to vary with temperature. Here, it can also be seen that the volume slightly increases the



(a) All values for the average volume per atom plotted against the temperature for all 97 materials; this is without the seven outliers.



(b) Same as Figure 16a, but then zoomed in and for a subset of materials.

Figure 16: Graphs with the average volume per atom (in Å^3/atom*) plotted against the temperature (in* K*). Each line represents a different simulated 2D perovskite material. The seven outliers as previously defined have already been removed.*

higher the temperature; this is to be expected, due to higher temperatures leading to the system having more energy, and thus, the atoms being able to, on average, move further away from their equilibrium positions. Some of the lines in Figure 16a, which is better visible in Figure 16b, show a sharp increase in volume at a certain temperature. This does not indicate an outlier, but rather the physical behaviour of the system, like thermal expansion or a phase transition.

Therefore, based on variation in values for the average volume per atom over each temperature and the stability of the average values, the average volume per atom is a suitable target to predict for ML models.

4.3 Feature Construction

First, before a model can be trained to predict the targets identified in Section 4.2, features are needed to train the model.

CIF Information

The goal of this research project is to see if simple machine learning models can be used to predict 2D perovskite properties based on the CIF information. Therefore, information found in the CIFs are first turned into features. The information from the CIFs that are used as features include the lattice parameters, namely the lengths of the edges of the unit cell, a, b and c and the angles between each of the edges, α, β and γ. The size information of the unit cell is also included, namely the volume of the unit cell, the amount of atoms in a single unit cell (see Section 4.4). The reason the volume is included in this case, is because the volume of the static structure might be different from the volume as determined in the simulations, seeing as, as mentioned in Section 3.6, the simulated boxes are assumed to be orthogonal, which is not always true. Lastly, the 0th, 1st, 2nd and 3rd joint

degree distributions of the unique spacer molecules are also included as features. It should be noted that these features are collinear; the lattice parameters, size information and joint degree distributions contain features correlated features. This in itself is not a problem, but it should be taken into account when building the ML models and analysing their performance.

Joint Degree Distribution

Ideally, each unique spacer molecule with relevant information like bonds, bond orders, angles, dihedrals and symmetries would be represented. However, a three-dimensional representation of the molecule can not be used as a feature for traditional ML models. To still represent the molecules as features, to a certain extent, joint degree distributions of the molecules are added; in this case, the zeroth, first, second and third joint degree distributions are added as features. The zeroth joint degree distribution represents the atom counts, for example "10 C-atoms; 16 H-atoms". The first joint degree distribution represents fragments of two neighbouring atoms, for example "8 C-H fragments" or "3 N-H fragments". This continues for the second and third joint degree distributions, where an example of the second and third degree would be "5 H-C-H fragments" and "3 H-C-C-H fragments" respectively. All fragments are counted in such a way that they are only counted once per atom composing them.

The joint degree distributions are counted with the help of an adjusted version of part of the pipeline code. Here, OpenBabel is utilised once more to determine connectivity of the spacer molecules. During determining the connectivity of all spacer molecules of all CIFs, some errors are displayed. These errors are either about the formatting of the CIFs, or the bond order, both of which are not utilised in this research, and can therefore be ignored.

4.4 Model Construction and Evaluation

This section first explains which models are used in this research project to predict the 2D perovskite properties, after which it describes how these models are trained, validated and evaluated.

Model Types

As gathered from Section 4.2, the goal of the model is to try and predict photoelectric-related material properties gathered from MD simulations, namely the potential-energy temperature derivative and the average volume per atom, using structure-based features, like information from the CIFs and the spacer molecules. The focus of the model should therefore be to capture the relationship between these static structure-based features and the dynamic material properties. Since these targets are continuous, rather than categorical, a model is needed that is capable of making continuous predictions. Besides that, since the features and targets are physically meaningful, the model should be interpretable. Moreover, due to the small amount of available data, the model should be able to handle few data points. Finally, the model has to remain stable despite noise that can be introduced during the MD simulations.

Several model types fit these criteria, namely linear regression and random forest. Both, linear regression and random forest models are suitable for learning continuous relationships. Linear regression is useful because it gives interpretable predictions and captures linear trends in the data. Random forest, on the other hand, is able to capture more complex behaviour, as well as handling non-linear relationships in the data.

As described in Section 2.3, linear regression is solved using ordinary least squares. In this approach, the weight of each feature, also called a coefficient, is determined by how that specific feature influences the model compared to the other features. If the amount of features, which there are in this case 68 of, approaches the amount of data points, in this case 104, the features do not provide enough independent information. Consequently, the ordinary least squares method becomes highly sensitive

to noise, making the model unstable. For this reason it is expected that ordinary least squares linear regression is not suitable due to the size of the features set, relative to the amount of data.

To still be able to use a form of least squares both the LASSO and Ridge models can be used. Both these models are forms of linear regression, which penalise the coefficients. LASSO shrinks the weights of some features to zero, while Ridge shrinks the weights of some features without excluding them entirely. It is therefore expected that these models will be able to better handle the amount of features relative to the amount of available data.

Therefore the models that are used to predict the targets are Ordinary Least Squares, LASSO, Ridge and Random Forest, where it is expected that the Ordinary Least Squares model will not perform well, but is still included to confirm these assumptions.

Target-Specific Modelling Strategies

The potential-energy temperature derivative target is determined by using all 16 temperatures. The 16 different temperature runs are used to calculate the slope of the potential energy as function of the temperature, meaning the predictions of the model are defined per material, rather than per temperature. Since 104 materials are included in the dataset, this results in a total of 104 data points for this target.

The average volume per atom, on the other hand, shows a spread between different temperatures within the same material. Therefore, the average volume per atom will be predicted per temperature, not per material, as with the potential-energy temperature derivative. This means each average volume at a certain temperature is its own data point. However, because different temperature runs per material are strongly correlated, data leakage can occur. To prevent data leakage, the training, validation and test splits are always made per material, and never per temperature. This ensures that all temperature runs of the same material are in the same split.

The average volume per atom will be predicted using two different approaches, namely building a model from scratch, and building a model based on the predictions of another model. One of these approaches entails training a model from scratch, using the temperature as an additional feature. Another approach is to first predict the mean average volume per atom per material, thus over all 16 temperature runs, and then using the predictions of that model as an input feature, as well as the temperature, for the second model. This provides insight as to whether performance is improved when separately determining the contribution of the material-specific influence on predicting the volume, before including temperature-related contributions.

Hyperparameter Optimisation & Cross-Fold Validation

Since the performance of the models is dependent on the choice of hyperparameters, a specific approach to hyperparameter optimisation is used. To find the optimal hyperparameters for the models trying to predict 2D perovskite properties, GridSearch, a form of hyperparameter optimisation, is used. GridSearch is used to optimise the α parameter for the LASSO and Ridge models, and the maximum tree depth, the minimum number of samples per leaf, and the number of considered features at each split for the Random Forest model. Since LASSO uses both coefficient penalisation as well as feature selection through shrinking the weights of the features to zero, the model is highly sensitive to the value of *alpha*. The search grid for α is therefore on a logarithmic scale, with values starting from -10, ending at 1, with a total of 40 steps. For Ridge, the search grid for α uses values starting at 0.1, ending at 200, also using 40 steps. For Random Forest, the maximum tree depth is chosen as either 4, 6 or 8, the minimum number of samples per leaf as either 2, 3 or 5, and the number of considered features at each split as either 0.3, 0.5 or 0.8. Because Random Forest models are usually less sensitive to hyperparameters than linear models which penalise coefficients, the hyperparameter optimisation is expected to not influence the Random Forest model as much as compared to the hyperparameter optimisation for LASSO and Ridge.

Table 1: Evaluation metrics for the Ordinary Least Squares, LASSO, Ridge and Random Forest models predicting the $C_p^{(\hat{U})}$. The metrics in this table are computed using the best found hyperparameters in each of the five folds. The Mean Generalisation Gap column displays the mean of the difference in validation and test performance over the five outer folds. The Test Set Mean R^2 column shows the average performance of the model across the five outer CV folds, and the Test Set Std. R^2 contains the corresponding standard deviation. In the NRMSE column, the normalised values of the root mean squared error, belonging to the predictions made by the model, can be found. The NRMSE is calculated from the test predictions from each of the five folds. Two outliers have been removed prior to training.

Model	**Mean Generalisation Gap**	**Test Set Mean R^2**	**Test Set Std. R^2**	**NRMSE**
Ordinary Least Squares	-751	-31.6	27.9	1.11
LASSO	0.014	0.219	0.110	0.166
Ridge	0.032	0.215	0.162	0.166
Random Forest	-0.016	0.494	0.177	0.134

Because hyperparameter optimisation is used, there needs to be a train, validation, test split of the data; not doing so would introduce data leakage. To do this, five folds are created which contain different splits in which there is 4/5 training data, and 1/5 test data; these folds are the outer folds. The outer folds are split based on the material, such that each of the 16 temperature runs for a single material always belong to the same split. Then, within each fold, the training data from the outer fold is split up into, 4/5 inner fold training data, and 1/5 validation data. The model is trained on the inner fold training data, after which the hyperparameters are optimised using GridSearch, based on the evaluation of the model on the inner validation split. A model with the hyperparameters giving the best performance on the validation set is then evaluated on the outer test data, which, at this point, is new data for the model. This is repeated for each of the five outer folds, in which each inner fold returns a model with the best hyperparameters based on the inner fold-specific validation split.

5 Results

This section shows how well the combination of the selected targets, described in Section 4.2, constructed features, described in Section 4.3, and the potential simple ML models, described in Section 4.4, is able predict to 2D perovskite properties based on features derived from static CIFs.

5.1 Average Potential-Energy Temperature Derivative ($C_p^{(\hat{U})}$) Models

The first target that is predicted is the potential-energy temperature derivative, as described in Section 4.2.1. Since the entire temperature range (200K - 350K) is used to find the derivative, there are as many data points as uniquely simulated materials, namely 104.

The models used in this research to try and predict the potential-energy temperature derivative are Ordinary Least Squares, LASSO, Ridge, and Random Forest. To find out which model is the most accurate in predicting the potential-energy temperature derivative from 2D perovskites, the metrics and hyperparameter optimisations, as described in Section 4.4, are used. These metrics include the mean generalisation gap, the outer five-fold Cross Validation mean R^2, the corresponding standard deviation of the R^2, and the NRMSE. These results can be found in Table 1.

Table 1 shows the different evaluation metrics for the four models. Higher values of R^2 indicate better predictions made by the model. The mean generalisation gap is used as an indication as to

Table 2: Comparison table for model performance for Ordinary Least Squares using all features, and Ordinary Least Squares using a small subset of features. There are only two features in said subset, namely the atom density, and the 0th degree distribution.

Model	**Mean Generalisation Gap**	**Test Set Mean R^2**	**Test Set Std. R^2**	**NRMSE**
Least Squares (All Features)	-751	-31.6	27.9	1.11
Least Squares (Two Features)	-0.074	0.160	0.094	0.173

how well the model generalises on unseen data. It is calculated by subtracting the outer-fold test R^2 from the inner-fold validation R^2, where the model adapts to the validation set by using it for hyperparameter tuning, while the test set remains unseen by the model. This means a more positive value indicates the model performing better on the validation data than the unseen data, which could be a sign of overfitting, whereas a more negative value indicates the model performs better on the test set than on the validation set, which could be a sign of an unlucky validation split. A value around zero would mean similar behaviour between the validation and test set, which is the desired outcome. The mean R^2 of the Test Set demonstrates how well the model generalises to the unseen test data from each of the outer five folds, while the standard deviation of R^2 shows how much the test set performance of the model differs across folds, and thus provides an indication of how consistent the model is between different train-test splits. The value for R^2 is in between $-\infty$ and 1; this is because the error of the model has no finite lower bound, hence $-\infty$, but the best the model can do is match the provided data, hence 1. This means a higher value for R^2 is desired. Finally, the NRMSE is included. The NRMSE is used, rather than the RMSE, since the RMSE is dependent on the scale of the target variable, and thus difficult to interpret. The NRMSE is a value relative to the scale of the target variable. The higher the value of the NRMSE, the greater the error of the prediction relative to the target's scale.

Ordinary Least Squares is included in Table 1. However, as expected and explained in Section 4.4, using Ordinary Least Squares resulted in extreme negative five-fold Cross Validation R^2 values and high NRMSE values. To confirm the expectation of this being due to the amount of features, the Ordinary Least Squares model was run once more using only 2 out of the 68 features; these results are shown in Table 2. Looking at Table 2 indicates that the worse performance of Ordinary Least Squares is due to the amount of features being close to the amount of used data points. The Ordinary Least Squares model is therefore not further discussed or included in the rest of the analysis, since reducing the amount of features only for the Ordinary Least Squares model makes it unreliable to compare to the other three models.

5.1.1 Analysis of Results

First, looking at the generalisation as well as the Test Set Mean and Standard Deviation R^2 values, shows varying results. Both, the LASSO and Ridge models perform similarly. While the generalisation gap for LASSO is slightly lower than for Ridge, both are close to 0. Although, for both models, the Test Set R^2 values are around 0.22, possibly showing that the captured relationship between the static CIF features and the target is weak. The standard deviation across the five outer folds is 0.052 higher for Ridge, as compared to LASSO. The difference could be due to how LASSO and Ridge handle the coefficients of the features: as explained in Section 4.4, LASSO removes features, while Ridge only changes the importance of certain features. It is possible certain features behave differently across folds, and since Ridge does not remove the influence of any features, it could lead to a higher standard deviation. The overall higher standard deviation for both LASSO and Ridge models could be due to the small dataset. On the other hand, the Random Forest model seems to perform better with a Test Set R^2 of 0.494 and a generalisation gap of -0.016, meaning the RF model is able to better capture

Table 3: Normalised mean distance and standard deviation of the predicted potential-energy temperature derivative data points of the LASSO, Ridge and Random Forest models, to the diagonal of the parity plots. The diagonal indicates the correct predictions.

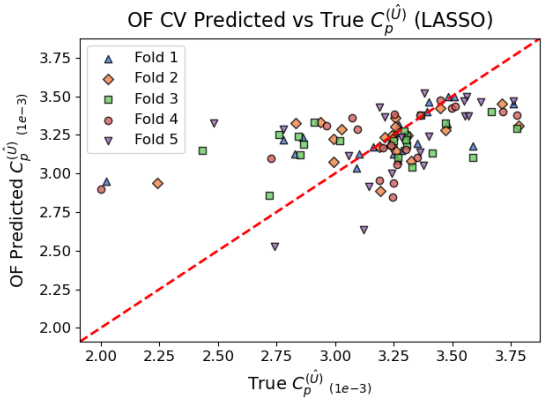
Model	**Normalised Mean Distance to $y = x$**	**Normalised Std. Distance to $y = x$**
LASSO	0.088	0.078
Ridge	0.089	0.076
Random Forest	0.070	0.064

behaviour in the data and generalises well, although the high standard deviation of 0.177 does suggest the model being less consistent across folds. The higher standard deviation could be due to the limited amount of data: the model is more likely to pick up on patterns stemming from noise in the training data, which the Random Forest tries to include in its splits, thereby capturing noisy patterns. The NRMSE values across the models are in line with this: LASSO and Ridge have the highest NRMSE, meaning it exhibits the largest prediction error, while Random Forest has the lowest NRMSE and thus performs the best. The reason the Random Forest model performs better than both the LASSO and Ridge models could be due to non-linear trends in the relationship between the features and the target, which are unable to be captured by forms of linear regression, like LASSO and Ridge. Alternatively, the relationship between the features and the target might be weak or noisy, or there might be too many, or redundant, features, which makes it more difficult for linear models to learn the behaviour of the target. However, the Test Set R^2 of the Random Forest model remains on the lower side, and thus is not able to capture the full behaviour of the target either. This implies that either the relationship between the feature and target is only partially learnable from the available data, or that the static CIF features are unable to convey the information required to capture the behaviour of the target. It should be noted that the hyperparameters of the outer folds sometimes differ between folds, but this is expected due to the limited size of the dataset. For example, for LASSO all but one are in the same order of magnitude, namely E-5, while for Ridge the two most extreme values are on opposite sides of the hyperparameter grid.

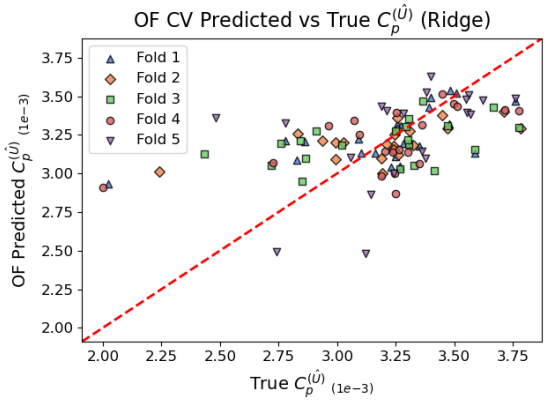
5.1.2 Prediction Error Analysis

To get an idea of how close the predicted values are compared to the true values, parity plots are used. These plots show the true value plotted on the horizontal axis and the predicted value on the vertical axis, where the ideal behaviour of the model is indicated by the $y = x$ line. Dots lying close to this line are a sign of accurate predictions. To evaluate the model's predictive capabilities, the used parity plots are constructed using test predictions from each of the five outer Cross-Validation folds. It is important to keep in mind that the test data originates from five different folds, meaning the predictions are based on five different models with possibly five different sets of hyperparameters.

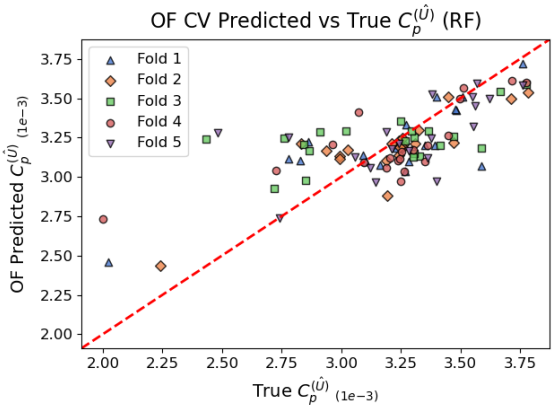
Figures 17a and 17b show the parity plots for the LASSO and Ridge models predicting the potential-energy temperature derivative. As expected from the information shown in Table 1, both LASSO and Ridge make similar predictions, with both of them making predictions closer to the mean, thereby not capturing the behaviour of the target at both ends of the diagonal. This can be confirmed by comparing the mean distance of all outer-fold test predictions to the diagonal line. Table 3 shows the normalised mean distances, as well as the standard deviation, to the diagonal $y = x$ for each of the three models. Here, it can be seen that, indeed, LASSO and Ridge have higher mean values as compared to the Random Forest model, although none of the three models entirely collapse towards the mean. The parity plot for the Ridge model seems to have more spread out predictions as opposed to the parity plot for the LASSO model, which could be due Ridge not reducing coefficients to zero like LASSO does; see Section 4.4.



(a) LASSO parity plot, predicting the $C_p^{(\hat{U})}$.



(b) Ridge parity plot, predicting the $C_p^{(\hat{U})}$.

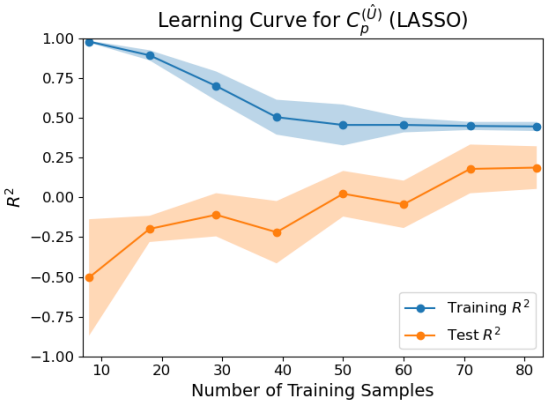


(c) Random Forest parity plot, predicting the $C_p^{(\hat{U})}$.

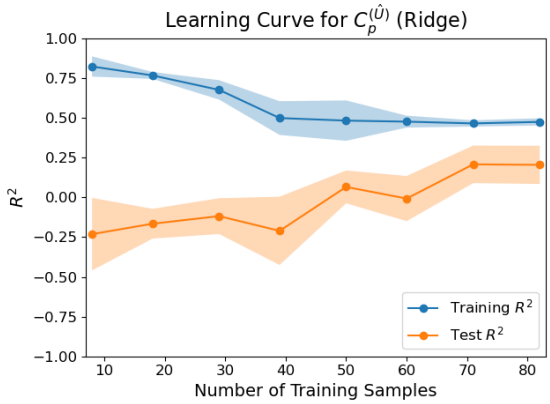
Figure 17: Parity plots for the LASSO, Ridge and Random Forest models, where the outer-fold predictions of $C_p^{(\hat{U})}$ are plotted against the true values of $C_p^{(\hat{U})}$. Since five-fold cross-validation is used, each of the outer-fold test predictions of each of the five folds is shown in this plot, and indicated with a unique colour and shape. The red-striped line in the middle shows the $y = x$ line, showing correct predictions. Note, two outliers were removed prior to training.

Figure 17c shows the parity plot for the Random Forest model predicting the potential-energy temperature derivative. The Random Forest model is able to capture more variation in the behaviour of the target, especially at the extremes of the diagonal—as opposed to the LASSO and Ridge models—as well as the predictions being less centred around the mean. This confirms the previous explanation of the Random Forest model being able to capture more of the feature-target relationship, but still not the complete behaviour of the target, since predictions still deviate from the diagonal.

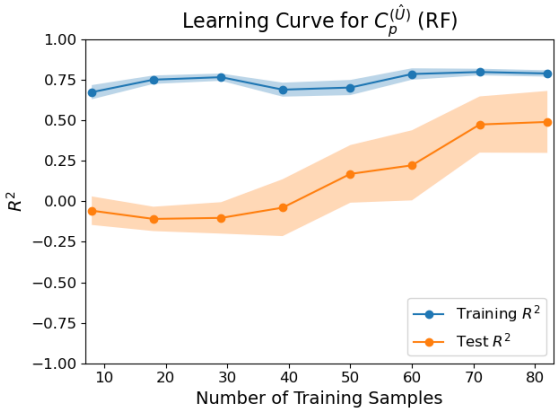
Although some of the hyperparameters between folds differ, Figure 17 seems to show a consistent distribution between the data points of each of the five folds, meaning the difference in hyperparameters of the five optimised models likely does not affect their predictive power much.



(a) Learning curves for LASSO predicting the $C_p^{(\hat{U})}$. The hyperparameter used for this model is $\alpha = 1.19 * 10^{-5}$.



(b) Learning curves for Ridge predicting the $C_p^{(\hat{U})}$. The hyperparameter used for this model is $\alpha = 20.8$.



(c) Learning curves for Random Forest predicting the $C_p^{(\hat{U})}$. The hyperparameters used for the RF model are a maximum forest depth of 4, a minimal amount of 2 leaves and a max features setting of 0.3.

Figure 18: Learning curves for the LASSO, Ridge and Random Forest models predicting the $C_p^{(\hat{U})}$, displaying the training R^2 performance (blue curve) and the corresponding test R^2 performance (orange curve) as a function of the amount of samples the model has been trained on. The train-test split is performed using five-fold cross-validation. The shaded areas around the curves visualise the standard deviation across folds. Note, two outliers were removed prior to training.

5.1.3 Learning Behaviour

Figures 18a and 18b visualise the learning curves of the LASSO and Ridge models. The learning curves are determined by training each of the models several times on increasing subsets of the available training data, after which the performance on the training and test sets is computed. Note that per training iteration the size of the training set changes, while the size of the test set remains the same. These curves can be seen as a separate experiment aiming to provide insight into the learning behaviour of the models. This means that the data splits used to generate the learning curves are

different from the ones used in, for example, the parity plots.

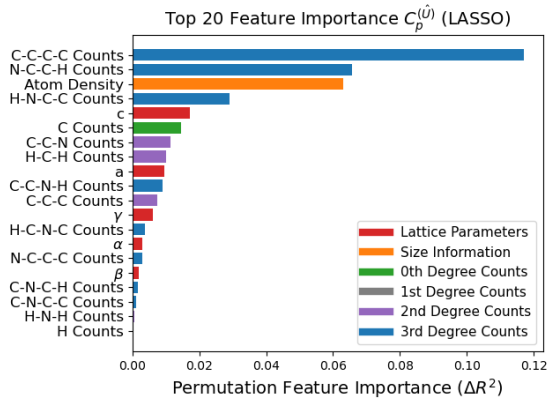
Both these models have similar learning curves, although the separation between the learning curves for the LASSO model is larger than for the Ridge model. For both models, the training set performance curve starts out high, meaning that at few samples the models are overfitting. At the same time, the test set performance curves start at negative values, confirming that the model is overfitting on the training data, and thus poorly generalising on the test data. As the model is trained on more samples, the training set performance curves decrease, and the test set performance curves increase, meaning the model becomes more generalisable. The test set performance curves for both models show an upward trend, meaning it is possible the model would have been able to make better predictions if more data was available. For both, the LASSO and Ridge models, no clear learning plateaus are present.

The learning curves for the Random Forest model, as shown in Figure 18c, show different behaviour. The training set performance curve does not start as high as compared to LASSO and Ridge, meaning the model does not overfit as much at smaller sample sizes. This is likely due to Random Forest being able to average predictions over decision trees, thereby improving the ability of the model to generalise at small sample sizes. The training set performance curve also stays around the same value throughout the training process, as opposed to the training set performance curves for LASSO and Ridge, which start out high and then end up lower. This could be, again, because the Random Forest uses multiple decision trees to average the predictions of the model. The test set performance curve does now start at around zero, and then continues to increase the more samples the model is trained on. For the LASSO and Ridge models, the standard deviation of the training split performance curves start out small, then become larger, and finally become small again, as opposed to the Random Forest model, which has a consistent small standard deviation for the training split performance curve. As Random Forest averages its predictions over the decision trees, the standard deviation for the training curve stays consistent. For LASSO and Ridge this is not the case: first, the sample size is small, leading to the same performance across folds, then, the sample size increases, meaning the data across folds becomes more variable, leading to a higher standard deviation, and finally the model is trained on all training examples, leading to similar data across folds on average. Noticeable in Figure 18c is that the spread of the test set performance curve increases as the sample size the model is trained on becomes larger, as opposed to Figures 18a and 18b where the spread of the test set performance curve remains similar throughout training. This is likely due to a variability in the created Random Forest models for each of the five folds. Because the predictions are also averaged over the decisions trees in the Random Forest, the standard deviation becomes larger the larger the sample size. On the other hand, the LASSO and Ridge models are based on the single α hyperparameter, and are therefore exactly the same across the five folds, meaning the standard deviation stays more consistent as the sample size increases. These figures also explain the reason why the Random Forest model has a higher test R^2 than LASSO or Ridge: since the Random Forest model is able to capture non-linear relationships, as well as better handle noise and co-linear features. The learning curves, and the conclusions drawn from them, are in line with the analysis of the parity plots.

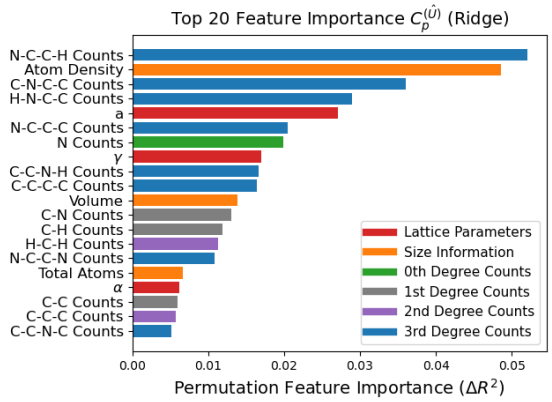
5.1.4 Feature Importance Analysis

Figures 19a, 19b and 19c show the Feature Importance of the models. Since the features are highly correlated, permutation-based feature importance is used. For the LASSO and Ridge models, the 3rd degree distribution of the molecules seems to be the most informative feature group. An explanation for this could be that the 3rd degree distribution, containing parts of three bonded atoms (like C-C-H) is the most important since it conveys the most information with regards to the composition of the spacer molecules. For the Random Forest model, the 2nd and 3rd degree distributions, as well as the lattice parameters, like the box angles and edges, seem to be important. Across the models, both the 0th and 1st degree distributions of the molecules do not seem to convey information used to learn the feature-target relationship.

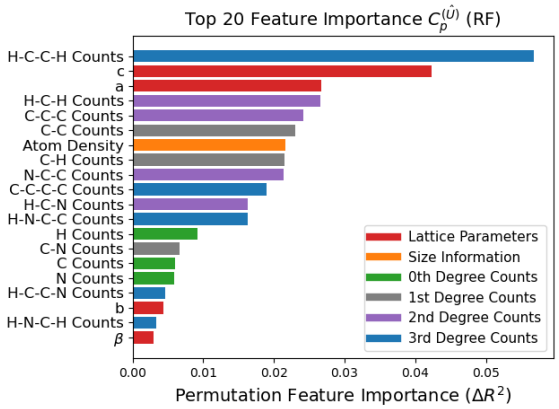
Looking at Figure 19c 2nd degree distribution seems to be an important feature group for the



(a) LASSO feature importances.



(b) Ridge feature importances.



(c) Random Forest feature importances.

Figure 19: Permutation-based feature importance plots for the LASSO, Ridge, and Random Forest models predicting the $C_p^{(\hat{U})}$. Feature importances are determined based on the performance on the outer-fold test sets, and show the top 20 features ranked from highest to lowest change in the test-set permutation performance. Each feature is coloured based on the group it belongs to. Note, two outliers were removed prior to training.

Random Forest model, while it seems to be one of the weaker feature groups for the LASSO and Ridge models, as seen in Figures 19a and 19b. This could be due to the Random Forest model being able to capture more complex relationships, like non-linear trends, between the 2nd degree distributions and the target, whereas the linear models are unable to do this.

Noticeable, however, is that most of the important joint degree distributions contain N-atoms. The N-atoms in the spacer molecules are usually formally charged and thus have a strong influence on how atoms locally interact and bond with each other. Besides, for all three models, the lattice parameter a also appears to be high, as well as c for the LASSO and Random Forest models. As discussed in Section 3.6, the simulations were ran as boxes, and not as slabs, because the identification of the directions in which the slab is supposed to expand could not be automated. However, upon manual inspection, most—but not all—materials seem to expand in the a and c directions. This is likely the reason these parameters are in the top 20 feature importance: the directions the slab expands in are influenced most by the temperature, since thermal expansion affects the directions the material

Table 4: Evaluation metrics for the Ordinary Least Squares, LASSO, Ridge and Random Forest models for predicting the $\hat{V}_{atom}$, using temperature as an input feature. Each temperature point (200K - 350K) for each material is used as a separate data point. All temperature points belonging to the same material are always kept within the same data split. The metrics in this table are computed using the best found hyperparameters in each of the five folds. The Mean Generalisation Gap column displays the mean of the difference in validation and test performance over the five outer folds. The Test Set Mean R^2 column shows the average performance of the model across the five outer CV folds, and the Test Set Std. R^2 contains the corresponding standard deviation. In the NRMSE column, the normalised values of the root mean squared error, belonging to the predictions made by the model, can be found. The NRMSE is calculated from the test predictions from each of the five folds. Seven outliers have been removed prior to training.

Model	**Mean Generalisation Gap**	**Test Set Mean R^2**	**Test Set Std. R^2**	**NRMSE**
Ordinary Least Squares	-208.2	-0.352	0.736	0.240
LASSO	-0.035	0.703	0.0622	0.108
Ridge	-0.185	0.635	0.153	0.116
Random Forest	0.003	0.789	0.114	0.095

expands in differently, with the in-plane directions, of the 2D perovskite, usually a or c, being more sensitive to temperature changes. Manual inspection of the CIFs shows a and c to usually be the directions in which the slab expands. Therefore, even though the determination of the directions in which the slab expands was not automated, it is expected to be a and c, which are exactly two of the model's top features.

5.2 Average Volume per Atom ($\hat{V}_{\mathbf{atom}}$) Models

The second target that is predicted is the average volume per atom, as described in Section 4.2.2. Now, each temperature, from 200K to 350K, for each material is used as a unique data point. This means that there are 104×16, which is 1664, unique data points. As mentioned in 4.2, it is important to keep in mind that each of the 16 temperatures of a single material are strongly correlated, and are therefore always included in the same split.

This section, elaborates on which of the used model families, namely Ordinary Least Squares, LASSO, Ridge and Random Forest, most accurately predict the target, which is, in this case, the average volume per atom.

Table 4 shows the different evaluation metrics for the four models. As expected, the Ordinary Least Squares model did not perform well, which is likely due to the size of the feature set relative to the amount of targets, as described in 2.3 and 4.4. Although the models are now trained on 16 times more data, this data is still strongly correlated. Therefore, as expected, the performance of the Ordinary Least Squares did not improve much, and thus will not be further discussed throughout this section.

5.2.1 Analysis of Results

Looking at Table 4 shows varying performance between the different models. First, looking at the mean generalisation gaps between the models, as a measure for how well the per-fold model with the best hyperparameters based on the validation set performs on the unseen test data. Here, the gap for the LASSO and Random Forest models are close to zero, meaning the models performs similar on both the validation and test sets. The generalisation gap for the Ridge model, however, is higher. This

Table 5: Normalised mean distance and standard deviation of the predicted data points for the average volume per atom of the LASSO, Ridge and Random Forest models, to the diagonal of the parity plots. The diagonal indicates the correct predictions.

Model	**Normalised Mean Distance to $y = x$**	**Normalised Std. Distance to $y = x$**
LASSO	0.053	0.055
Ridge	0.051	0.065
Random Forest	0.044	0.060

is a likely indicating of the Ridge model being more sensitive to the chosen hyperparameters during validation. This could be due to the search grid for the hyperparameters being too small, which could be made worse due to the strongly correlated features in the dataset.

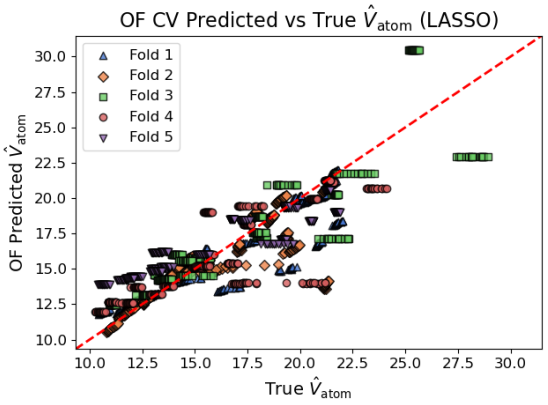
The Test Set Mean R^2 for LASSO and Ridge are different in performance, with LASSO having a value of 0.703 and Ridge a value of 0.635. The Test Set standard deviation for LASSO is lower than for Ridge. Since LASSO reduces coefficients to zero, it might reduce the influence of noise introduced by the correlated features. Since Ridge uses all features, it is possible it has more trouble filtering out the noise. Table 4 shows that the Random Forest achieves the highest Test Set R^2 value, outperforming both the LASSO and Ridge models. Likely, since Random Forest uses many decision trees to make its predictions, it is better able to suppress noise, handle collinear and redundant features, and find non-linear relationships, thus leading to a better overall performance compared to Ridge and LASSO. The higher standard deviations for all three models are probably due to the smaller dataset, which allows noise to have a stronger influence on predictions, and thereby hinders the models ability to better learn the feature-target relationship.

The NRMSE values for all three models confirm this: the NRMSE values are similar, with LASSO and Ridge having values closer to each other, and the Random Forest model having a lower NRMSE. Overall, these NRMSE values are high, which, again, could be explained due to the smaller dataset making it more difficult for the models to learn the relations between the features and the target, as well as resulting in less variation in volume across data points.

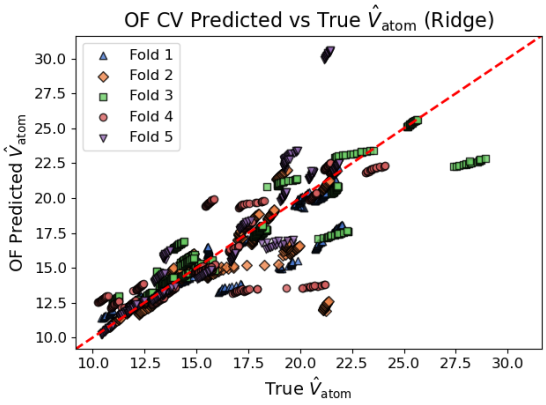
5.2.2 Prediction Error Analysis

The parity plots as shown in Figure 20 visualise how close the predicted values are compared to the true values. To evaluate the model's predictive capabilities, the used parity plots are, again, constructed using test predictions from each of the five outer Cross-Validation folds, where each of the data points is only used once in the test splits. The parity plots now contain more data points due to each of the 16 temperature simulations per material are considered to be a unique data point.

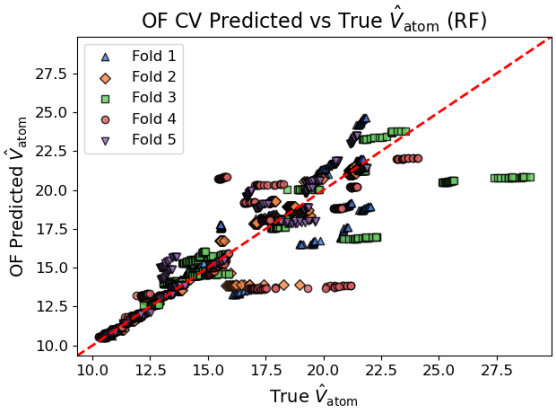
Figures 20a and 20b show the parity plots for the LASSO and Ridge models respectively. Ridge seems to have more accurate predictions than LASSO at higher volumes (from around 22.5), although it also appears that the predictions of the Ridge model have a wider overall spread compared to LASSO. Tables 4 and 5 confirm this; the Test Set R^2 value for LASSO is higher than that for the Ridge model, and Ridge has a higher mean generalisation gap, which can be explained by the larger spread in predictions. Looking at Figure 20c shows the Random Forest model having more accurate predictions at lower and medium volumes and a lesser spread in the overall predictions. This is in line with Table 4, indicating the Random Forest model makes the best predictions. It is important to note that the clusters of volumes specific to some of the materials deviate from the diagonal for all three of the models. This could be an indicating of the models failing to capture part of the feature-target relation. It is also possible that structures with an overall higher volume are more difficult to predict, either due to those being more uncommon in the dataset, or less predictable overall since structures with a higher volume might be influenced more by longer-range structural effects, that the current



(a) LASSO parity plot, predicting the $\hat{V}_{atom}$.



(b) Ridge parity plot, predicting the $\hat{V}_{atom}$.



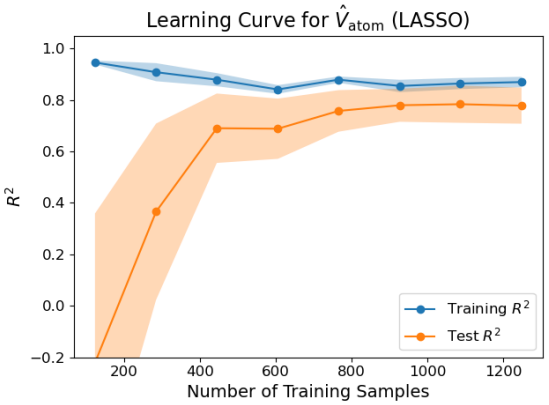
(c) Random Forest parity plot, predicting the $\hat{V}_{atom}$.

Figure 20: Parity plots for the LASSO, Ridge and Random Forest models, where the outer-fold (test set) predictions of $\hat{V}_{atom}$ are plotted against the true values of $\hat{V}_{atom}$. Since five-fold cross-validation is used, each of the outer-fold test predictions of each of the five folds is shown in this plot, and indicated with a unique colour and shape. The red-striped line in the middle shows the $y = x$ line, showing correct predictions. Note, seven outliers were removed prior to training.

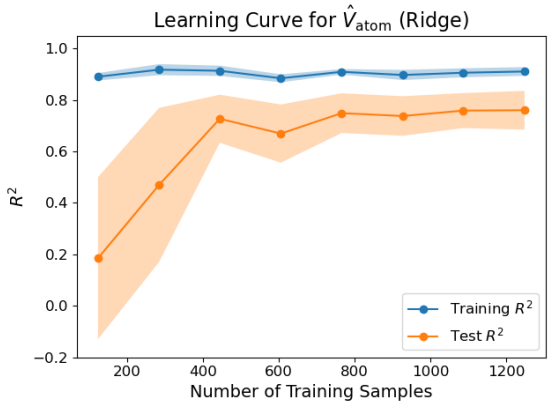
features do not fully describe.

5.2.3 Learning Behaviour

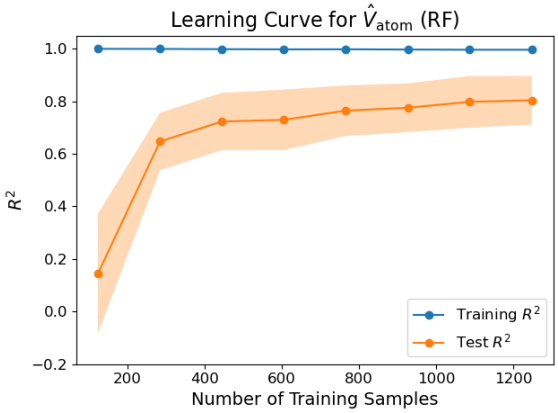
The learning curves for the LASSO, Ridge and Random Forest models can be found in Figure 21. Looking at Figure 21 shows the blue training R^2 curves at the top of the graphs. For the LASSO model, as can be seen in 21a, the curve starts out high, but then gradually decreases over time. Likely the model overfits at low sample sizes, but becomes more generalisable once it is trained on a larger variety of data. The training set curve for the Ridge model, as is visible in 21b, does not start out as high and displays a smaller decline as the number of training samples increases. Since Ridge does not zero out coefficients, the Ridge model has a overall high but slightly decreasing training R^2 as the sample size grows. Looking at the test set learning curve for the LASSO model shows severe underfitting at lower sample sizes. Since the average volume per atom is dependent on structural descriptors, and LASSO forces the coefficients of many features to zero, it severely underfits at lower sample sizes.



(a) Learning curves for LASSO predicting the $\hat{V}_{atom}$. The hyperparameter used for this model is $\alpha = 0.203$.



(b) Learning curves for Ridge predicting the $\hat{V}_{atom}$. The hyperparameter used for this model is $\alpha = 200$.



(c) Learning curves for Random Forest predicting the $\hat{V}_{atom}$. The hyperparameters used for the RF model are a maximum forest depth of 6, a minimal amount of 2 leaves and a max features setting of 0.3.

Figure 21: Learning curves for the LASSO, Ridge and Random Forest models predicting the $\hat{V}_{atom}$, displaying the training R^2 performance (blue curve) and the corresponding test R^2 performance (orange curve) as a function of the amount of samples the model has been trained on. The train-test split is performed using five-fold cross-validation. The shaded areas around the curves visualise the standard deviation across folds. Note, two outliers were removed prior to training.

As the number of training samples increases, the model is able to better identify physically relevant structural features, leading to improved predictive performance. This is also reflected in the standard deviation of the test curve, which starts out large but gradually becomes smaller. The test curve for the Ridge model shows similar behaviour, but on a lesser scale: the model initially shows underfitting, but since Ridge does not force coefficients to zero, the effect is not as severe as for LASSO. For both the LASSO and Ridge test curves, as the number of training samples increases, the test R^2 rises up until about 400 training samples, after which the test curve, especially for Ridge, appears to reach a plateau.

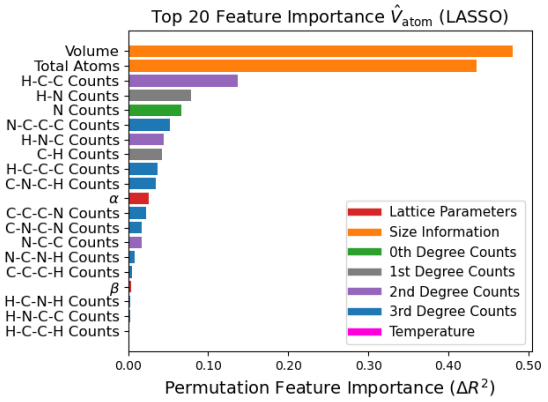
Looking at training set curve for the Random Forest model in Figure 21c shows the training R^2 consistently at around 1.0. The high training R^2 is likely due to the lack of variation between data points of the same material, leading to the model memorising the training data and fitting noise, rather than learning the feature-target relationship. However, the test R^2 curve shows a sharp rise of the R^2 at a low sample size. This shows that the Random Forest model is able to learn the main trends of the behaviour of the target, although it does not fully generalise and is likely limited by the structural features gathered from the CIFs. This is in line with the parity plot for the Random Forest, as seen in 20c, which shows the Random Forest model capturing the overall behaviour of the data well, while also displaying deviations and a large spread in the predictions at higher volumes.

The learning curves show that all three models are likely overfitting on the training data. Although the predictions of the models appear to have a high testing R^2 value, it does seem like all three models are unable to learn the full relationship between the structural descriptors from the CIFs and the average volume per atom of the materials, indicated by the plateaus. This is not necessarily a sign of the models failing, but rather a limitation caused by the limited information captured by the structural features, and possibly the small size of the dataset as well.

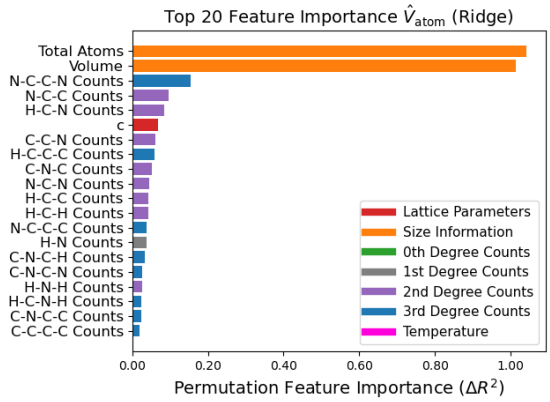
5.2.4 Feature Importance Analysis

Figure 22 shows the permutation-based feature importances for the LASSO, Ridge and Random Forest models predicting the $\hat{V}_{\text{atom}}$. Figures 22a and 22b show the Size Information feature group being important, with the *Volume* and *Total Atoms* features being the most important for the LASSO and Ridge models. This is to be expected since the static volume of the unit cell, and the amount of atoms in a unit cell, directly contribute to the average volume per atom and its behaviour under thermal expansion. These features likely explain simple changes in the size of the unit cell with temperature. However, unlike the LASSO and Ridge models, the Random Forest model has the Size Information as one of the weakest feature groups, as can be seen in Figure 22c; it does not have the *Volume* or *Total Atoms* feature in the top 20 important features as the LASSO and Ridge models. This indicates the Random Forest model is able to better capture the behaviour of the target through information conveyed by other features, like the joint degree distributions and lattice parameters, without needing to depend directly on the size information form the static CIF structure. Noticeable is that the *N Counts* feature contributes a lot more compared to any other features. The reason for this could be because the N-atoms likely convey important information regarding the spacer molecules, since they usually determine the total charge of the spacer molecule, and therefore strongly influence properties like the size, geometry, and ordering of the spacer molecules in the material. The random forest is able to use this seeing as it can discover and use non-linear relationships that cannot be fully captured by linear models. This is also in line with the feature importances for LASSO and Ridge, since most of the important joint degree distributions do contain N-atoms.

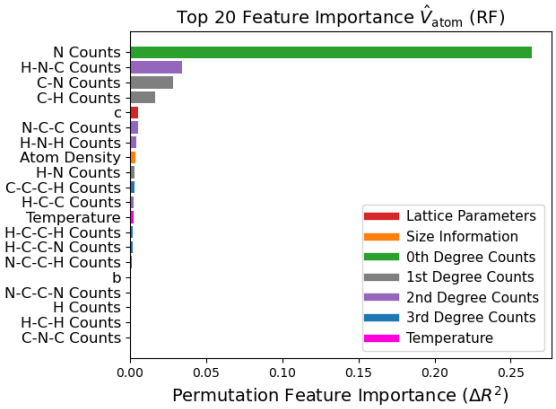
It is noticeable that the Temperature feature is an overall weak feature for all three models. A possible explanation is that the behaviour of the average volume per atom is conveyed through other features, like the Lattice Parameters and Size Information feature groups, rather than temperature itself as a feature. It is also possible that, due to the limited size of the dataset, the models did not have enough information to directly learn the behaviour of the target through the Temperature feature, and thus used other features to learn the behaviour.



(a) LASSO feature importances.



(b) Ridge feature importances.



(c) Random Forest feature importances.

Figure 22: Permutation-based feature importance plots for the LASSO, Ridge, and Random Forest models predicting the $\hat{V}_{atom}$. Feature importances are determined based on the performance on the outer-fold test sets, and show the top 20 features ranked from highest to lowest change in the test-set permutation performance. Each feature is coloured based on the group it belongs to, as shown in the legend. Note, seven outliers were removed prior to training.

5.2.5 Investigating Impact of Temperature Information

Based on the results shown in Section 5.2, the Random Forest model is able to best learn the behaviour of the target, based on the given features. This section therefore only considers the Random Forest model.

In this section, for the same target, namely the average volume per atom, two different models are constructed. First, the per-temperature model attempts to predict the target directly using the temperature as a feature; this is the same model as shown in the previous section. The stacked per-temperature model builds on this approach by using both the temperature and the predictions of a per-material focussed model as input features. For clarity, the stacked per-temperature model is referred to as the stacked model. The performance of the per-material model is not considered since it predicts the target at a per-material level, and therefore can not be compared to the other two models predicting on a per-temperature level. The per-material target is calculated by taking the average of all the average volume per atom values for each of the 16 temperatures. The comparison between

Table 6: Evaluation metrics for the per-temperature and stacked models Random Forest models predicting the $\hat{V}_{atom}$, using temperature as an input feature. Each temperature point (200K - 350K) for each material is used as a separate data point. All temperature points belonging to the same material are always kept within the same data split. The metrics in this table are computed using the best found hyperparameters in each of the five folds. The Mean Generalisation Gap column displays the mean of the difference in validation and test performance over the five outer folds. The Test Set Mean R^2 column shows the average performance of the model across the five outer CV folds, and the Test Set Std. R^2 contains the corresponding standard deviation. In the NRMSE column, the normalised values of the root mean squared error, belonging to the predictions made by the model, can be found. The NRMSE is calculated from the test predictions from each of the five folds. Seven outliers have been removed prior to training.

Model type	**Mean Generalisation Gap**	**Test Set Mean R^2**	**Test Set Std. R^2**	**NRMSE**
Random Forest: Per-Temperature	-0.062	0.748	0.111	0.105
Random Forest: Stacked	0.112	0.727	0.114	0.109

the per-temperature and stacked temperature models might reveal whether adding information on a per-material basis improves the models ability to learn the feature-target relationship.

Prediction Error Analysis

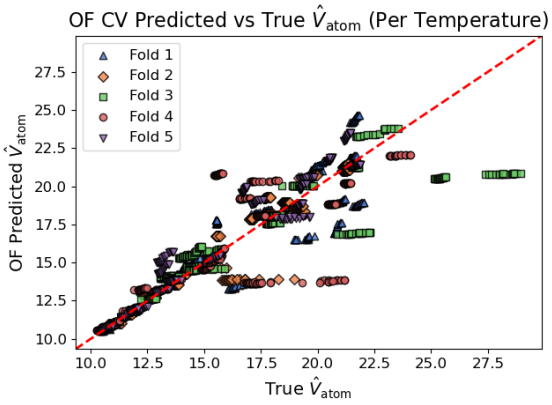
Table 6 shows the performance metrics for the per-temperature and stacked Random Forest models predicting the $\hat{V}_{\text{atom}}$ target. First, looking at the Mean generalisation Gap shows the value for the per-temperature model being negative, but near zero. This indicates the validation performance is similar to the test performance, meaning the best found hyperparameters generalise well to unseen data, which is the desired outcome. The stacked model on the other hand has a higher positive value, indicating the model overfits on the validation set. The Test Set R^2 for both the per-temperature and stacked models, as well as the standard deviation, is similar; the per-temperature model is able to better learn the behaviour of the target, based on unseen data than the stacked model, although not by much. The same can be said for the NRMSE, with both models performing similarly.

Prediction Error Analysis

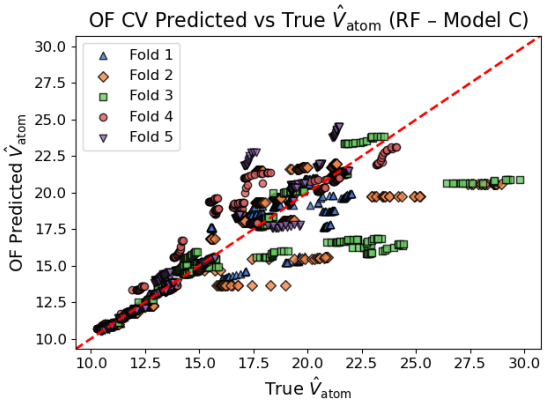
Figure 23 shows the parity plots for the per-temperature and stacked models. Figures 23a and 23b show similar predictions. The per-temperature model seems to make better predictions towards the middle of the graph, although the overall spread between the per-temperature and stacked models looks similar. This is to be expected when looking at Table 6, which displays the Test Set R^2 of the two models to be very similar, with the per-temperature model having a slightly higher value than the stacked model. The table showing the normalised mean distance, and the standard deviation, to the diagonal are not shown, since it is clear from Figure 23 the predictions are similar, and thus such information does not reveal anything new.

Learning Behaviour

The learning curves for the per-temperature and stacked models are not used to measure performance. Since the stacking approach used for the stacked model uses the predictions from the per-material model, the stacked model has access to the predictions of the per-material model. However, when creating the learning curve, the amount of training data the stacked model has access to is varied, while the stacked model still has full access to the predictions from the per-material model, which can introduce data leakage. Specifically, the per-material predictions used by the stacked model may have been generated using information outside the training set, allowing information from the

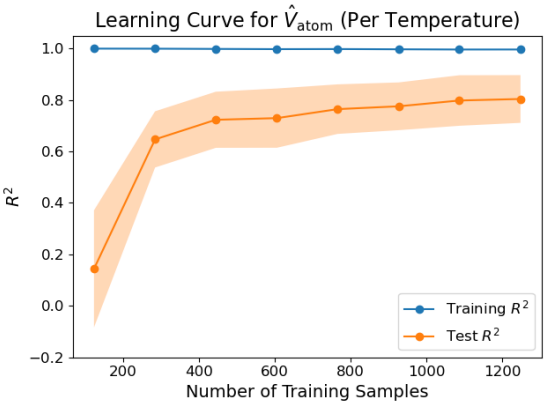


(a) Random Forest per-temperature model parity plot, predicting the $\hat{V}_{atom}$.

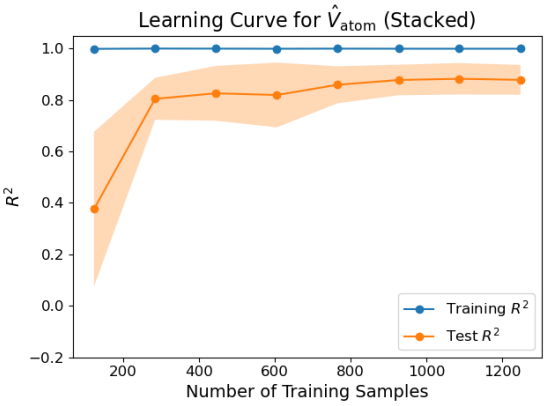


(b) Random Forest stacked model parity plot, predicting the $\hat{V}_{atom}$.

Figure 23: Parity plots for the per-temperature and stacked Random Forest models, where the outer-fold predictions of the $\hat{V}_{atom}$ are plotted against the true values of the $\hat{V}_{atom}$. Since five-fold cross-validation is used, each of the outer-fold test predictions of each of the five folds is shown in this plot, and indicated with a unique colour and shape. The red-striped line in the middle shows the $y = x$ line, showing correct predictions. Note, seven outliers were removed prior to training.



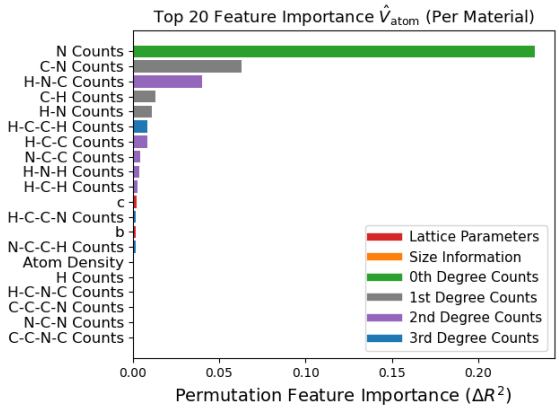
(a) Learning curves for the per-temperature Random Forest model predicting the $\hat{V}_{atom}$.



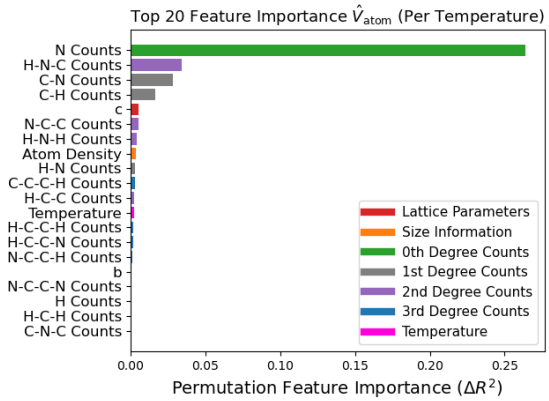
(b) Learning curves for the stacked Random Forest model predicting the $\hat{V}_{atom}$.

Figure 24: Learning curves for the per-temperature and stacked Random Forest models predicting the $\hat{V}_{atom}$, displaying the training R^2 performance (blue curve) and the corresponding test R^2 performance (orange curve) as a function of the amount of samples the model has been trained on. The train-test split is performed using five-fold cross-validation. The shaded areas around the curves visualise the standard deviation across folds. Note, two outliers were removed prior to training.

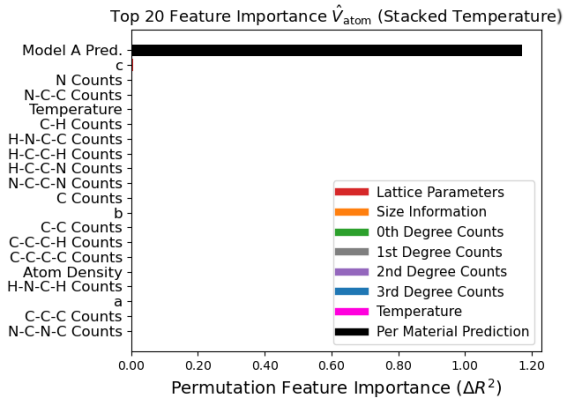
full dataset to leak into the stacked model during training, thus making it seem the stacked model performs better. This data leakage effect is observed for the stacked model used in this research, which cannot be avoided due to limitations of the used packages. For clarity, the learning curves of both the per-temperature and stacked models can be seen in Figures 24a and 24b. Since the stacked model outperforming the per-temperature model is not in line with with Table 6 and the parity plots of the per-temperature and stacked models as shown in Figure 23, the stacked model's learning curve



(a) Random Forest feature importances, based on the per-material model predicting $\hat{V}_{atom}$.



(b) Random Forest feature importances, based on the per-temperature model predicting $\hat{V}_{atom}$.



(c) Random Forest feature importances, based on the stacked model predicting $\hat{V}_{atom}$.

Figure 25: Permutation-based feature importance plots the per-temperature and stacked Random Forest models predicting the $\hat{V}_{atom}$. Feature importances are determined based on the performance on the outer-fold test sets, and show the top 20 features ranked from highest to lowest change in the test-set permutation performance. Each feature is coloured based on the group it belongs to, as shown in the legend.

is assumed to be the result of data leakage, and is therefore considered unreliable.

Feature Importance

Figure 25 shows the feature importances for the per-material, per-temperature and stacked Random Forest models, where the feature importance for the stacked model, 25c, has the addition of the *Per Material Prediction* feature, as coloured in black. Comparing Figures 25b and 25c shows that the only important feature for the stacked model are the predictions from the per-material model. Since the per-material model predicts the per-material $\hat{V}_{\text{atom}}$ average of the same target as the stacked model, the stacked model seems to capture almost no additional information with the addition of the Temperature feature; this effect is emphasised by the permutation-based approach. It is therefore expected that the per-material feature importance is similar to the per-temperature feature importance, which can be confirmed by looking at Figures 25a and 25b. These results show that the stacked approach does

not improve the ability of the model to learn the behaviour of the target using the temperature, based on the average $\hat{V}_{\text{atom}}$ per material behaviour.

6 Discussion

Section 5 shows that, depending on the target and the model family, the machine learning models are able to learn part of the behaviour of the targets. The behaviour of the $C_p^{(\hat{U})}$ was more difficult to learn for the models than the $\hat{V}_{\text{atom}}$, which is likely due to the volume being more directly related to the structural features of the material, whereas the $C_p^{(\hat{U})}$ is not as directly related to any of the features and also takes into account the thermal behaviour in the material, which is not directly represented by any of the features.

The $C_p^{(\hat{U})}$ being more difficult to predict could also be due to the target being the potential-energy temperature slope: since the target is based on 16 temperature simulations per material, determining the average value for the potential-energy per temperature introduces noise, and then using those 16 points to find the slope of the $C_p^{(\hat{U})}$ introduces more noise; this is likely worsened due to the small dataset.

Predicting the $\hat{V}_{\text{atom}}$ does not seem to suffer as much from a limited dataset, although there might be a stronger relationship between the target and the temperature, which, in this case, was not found. For the results found in this research, the temperature does not seem to convey much information that the models use to learn the feature-target relationship. It is therefore likely that, when wanting to do predictions with regards to the $\hat{V}_{\text{atom}}$, a single simulation at a certain temperature can be ran, rather than needing all 16 temperatures. If this is the case, more computational power can go towards extending the duration of the simulation, as well as the size of the unit cell in the simulations, so to increase the chemical correctness of the simulations. However, for predicting the $C_p^{(\hat{U})}$, a temperature range is still required.

Chemical Relevance

Molecular Dynamics simulations are a useful tool to computationally simulate and examine structures of materials without having to experimentally create them. However, since this is done computationally rather than using physical samples, it raises the question of whether the simulations of such a computational approach are able to maintain chemical correctness and accurately represent the chemically relevant aspects of the structures. This is especially relevant since the decision was made to run the 2D perovskite simulations in orthogonal simulation boxes, rather than as slabs. It is therefore important to consider whether the results found during this research, apart from their machine learning aspect, also make sense from a chemical point of view.

First, the feature importances of the models showed nitrogen (N-) atoms being among the most important joint degree distributions. The N-atoms in the spacer molecules carry the role of formally charged atoms, and therefore influence how the atoms interact and bond with each other locally. As a result, the presence of the nitrogen atoms in the feature importances shows that both, the simulations, and the machine learning models, capture chemically meaningful interactions.

Furthermore, the lattice parameters also seem to be relevant with regards to the feature importances, especially the a and c parameters, which correspond to the lengths of the edges of the unit cell. Due to difficulties in automating the directions along which the 2D perovskite slabs are supposed to extend, the simulations were performed using orthogonal boxes. It was assumed that this would not influence the results as much since the unit cell was replicated in all directions, thereby minimising the influence of the use of orthogonal boxes, rather than slabs, on the behaviour of the simulated structures. Upon manual inspection, it seems like most of the unit cells of the 2D perovskites should be extended along the a and c directions. The fact that specifically the a and c lattice parameters are important for the machine learning models, and not b, therefore indicates that the models were able

to learn part of the behaviour of two 2D perovskite properties, despite using orthogonal boxes rather than slabs.

Lastly, the machine learning models appear to have more difficulties with predicting the potential-energy temperature slope, rather than the average volume per atom. This can be explained by the fact that the $C_p^{(\hat{U})}$ was constructed by first averaging the potential energy at each temperature, and then fitting a line to the data points over 16 temperatures; each of those steps introduces noise, making the target more difficult to predict. Additionally. the potential-energy temperature derivative describes the thermal behaviour of the system, which makes it not as much related to the static structural features gathered from the CIFs as the average volume per atom, and thus more difficult to learn. Although the models are not able to capture the full behaviour of the potential-energy temperature derivative, it still appears to capture trends that explain part of the physical behaviour of the system.

All together, these results indicate that the automated pipeline converting structural information from CIFs to simulation-ready data is not only useful for machine learning, but also chemically relevant.

The focus of this research was to find out how effective simple machine learning models are in accurately predicting the structural properties of 2D perovskites, directly from the data from CIFs. As shown throughout this research paper, the machine learning models are able to learn chemically meaningful relations between the static structural features from the CIFs, and the structural properties of the 2D perovskites, although the accuracy of the predictions depends on the target. Targets more closely related to the structural features, like the $\hat{V}_{\text{atom}}$, are easier to predict than targets which are also related to the thermal properties of the materials, like the $C_p^{(\hat{U})}$. These results show that simple machine learning models are indeed able to predict structural properties of 2D perovskites to an extent. However, the predictions of these models are held back by limited number of varying features from the static CIFs; this is emphasised by the existence of several learning curve plateaus. The goal of this research was, however, not to develop machine learning models that can differentiate between all unique structures, but rather to create models that are able to find subsets of 2D perovskite materials based on certain desired properties; a task which the models as presented in this research paper seem to be capable of performing. While the predicted properties used during this research are not optoelectric properties themselves, the predicted targets are still related to structural and thermal behaviour of the 2D perovskite materials, and therefore indirectly related to their optoelectric properties. The targets presented in this research paper therefore allow for the selection of subsets of 2D perovskite materials which can be used for further optoelectric research.

Limitations & Assumptions

There are, however, also some limitations with regards to the approaches used and results found during this research.

First, the Molecular Dynamics simulations are based on the CHARMM force field, alongside parameters taken from previous research. Ideally, force fields would be created specifically for each unique material, together with the corresponding parameters. However, developing and parametrising force fields is difficult and time consuming, making such an approach not viable for the scope of this research. Therefore a general force field is used in this research. Despite the force field not being specific to each material and using parameters derived from a previous study, the models still seem to learn trends describing the behaviour of the target, as well as seeming to be chemically consistent. This indicates that the decision to use a general force field did not prevent the models from learning chemically relevant trends.

Moreover, finite boxes and a limited simulation time were used during the simulations. The larger the replications of the unit cell, and the longer the duration of the simulations, the more computationally expensive the simulations will be. While the smaller size of the simulation box (3x3x3) as well as the chosen simulation time seem to be good enough for the machine learning models to find relations in the data, it is possible a larger simulation box and an extended simulation

time could improve the predictions of the models. It should be noted that, for this research, the simulations were more computationally expensive due to the inclusion of the 16 temperature ranges. As shown in this research, it is unlikely all 16 temperatures are needed; reducing the amount of temperatures would also save on computational costs.

Besides, as mentioned before, the simulations were run using orthogonal boxes, rather than simulation slabs and including the angles of the boxes in the simulations. Even though the models are able to learn part of the behaviour of the targets, it is possible that part of the structural behaviour of the 2D perovskites can not be fully captured by the models. Nonetheless, for both targets the models seem to be capturing trends describing the behaviour of the targets.

Lastly, another difficulty is the models predicting the potential-energy temperature derivative being limited in their predictive power due to the lack of features describing the thermal behaviour of the system. Since CIFs do not contain information that directly describe thermal behaviour, this is a limitation with regards to the available features in the CIFs, rather than the approach used during this research. Since this information is not easily obtainable—without first running the simulations—this limitation is largely unavoidable, but nevertheless hinders the model's ability to fully learn the behaviour of the target.

Future Research

Any future research that would be expanding on this work should focus on further improving and ensuring chemical validity, in both the simulations and the machine learning models.

From a simulation perspective, this can be done by improving the calculation of the charges, using larger simulation boxes, extending the duration of the simulations, simulating the 2D perovskites as slabs and including the lattice angles, rather than the simulation boxes being orthogonal, and using a force field that more accurately simulates organic-inorganic materials. Lastly, computational power can be saved by performing the simulations over a smaller temperature range, since this research has shown that the temperature does not meaningfully influence the predictions of the models

From a modelling point of view, the predictive power of the models, especially for $C_p^{(\hat{U})}$, could likely be improved if the thermodynamic and long-range properties of the systems are included in the features. Besides, a more compact set of features could improve the predictions and generalisability of the models. Lastly, by training the models on more data by expanding the 2D perovskite dataset, the predictions of the models could be more accurate.

7 Conclusion

The goal of this research is to find out how effective simple machine learning models are in accurately predicting the structural properties of 2D perovskites directly from CIFs, with the aim of identifying subsets of 2D perovskite materials with unique structural properties. Such a goal is useful for focusing on a select group of 2D perovskite materials relevant for further research, like constructing solar cells needing to withstand a specific temperature range using 2D perovskites.

To achieve this goal, first an automated pipeline was created to extract and convert structural information of the 2D perovskite CIFs to simulation-ready files. After this, the Molecular Dynamics simulations were used to simulate the 2D perovskite materials at varying temperatures. Finally, machine learning models trained on the structural information from CIFs were used to predict 2D perovskite properties based on the data gathered from the simulations.

Based on the results found during this research project, the simple machine learning models seem to be able to capture meaningful trends and learn the behaviour of the target properties to an extent—especially the Random Forest-based models. Despite the models not learning the full behaviour of the target, this was not the goal of this research; the goal is to differentiate between subsets of 2D perovskite materials, which the created machine learning models seem to be capable of. This would mean that selecting subsets of materials based on certain properties, with the intention to filter out

irrelevant structures without having to experimentally construct them or run MD simulations for all materials and further research the subset of materials, seems to be possible.

However, the approach taken in this research required several simplifications. First, during this research, a general force field is used for the Molecular Dynamics simulations. Besides, the features of the machine learning models are based on a small dataset containing only static structural information, most of which are correlated. Despite these limitations the results show that the approach is able to capture the behaviour of, chemically meaningful, trends, and can identify subsets of 2D perovskites. This confirms that the used approach is effective for the identification of subsets of relevant materials.

References

(1) Jin, L.; Duan, K.; Tang, X. What Is the Relationship between Technological Innovation and Energy Consumption? Empirical Analysis Based on Provincial Panel Data from China. *Sustainability* **2018**, *10*, DOI: 10.3390/su10010145.

(2) Deshmukh, M. K. G.; Sameeroddin, M.; Abdul, D.; Abdul Sattar, M. Renewable energy in the 21st century: A review. *Materials Today: Proceedings* **2023**, *80*, 1756–1759, DOI: https://doi.org/10.1016/j.matpr.2021.05.501.

(3) Chen, Y.; Sun, Y.; Peng, J.; Tang, J.; Zheng, K.; Liang, Z. 2D Ruddlesden–Popper Perovskites for Optoelectronics. *Advanced Materials* **2018**, *30*, DOI: 10.1002/adma.201703487.

(4) Zhao, X.; Ball, M. L.; Kakekhani, A.; Liu, T.; Rappe, A. M.; Loo, Y. L. A charge transfer framework that describes supramolecular interactions governing structure and properties of 2D perovskites. *Nature Communications* **2022**, *13*, DOI: 10.1038/s41467-022-31567-y.

(5) Thompson, A. P.; Aktulga, H. M.; Berger, R.; Bolintineanu, D. S.; Brown, W. M.; Crozier, P. S.; in 't Veld, P. J.; Kohlmeyer, A.; Moore, S. G.; Nguyen, T. D.; Shan, R.; Stevens, M. J.; Tranchida, J.; Trott, C.; Plimpton, S. J. LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Comp. Phys. Comm.* **2022**, *271*, 108171, DOI: 10.1016/j.cpc.2021.108171.

(6) Assis, A.; Dantas, J.; Andrade, E. The performance-interpretability trade-off: a comparative study of machine learning models. *Journal of Reliable Intelligent Environments* **2024**, *11*, 1, DOI: 10.1007/s40860-024-00240-0.

(7) Auyeung, C., *Unit Cells*; CK-12 Foundation: 2013; Chapter 13.14, p 13.14.

(8) Tilley, R. J. D., *Crystals and crystal structures*, 1st ed.; John Wiley: 2007, p 255.

(9) Mihaly, L.; Martin, M. C., *Solid State Physics: Problems and Solutions*, 1st ed.; John Wiley: 1996, p 261.

(10) Oganov, A. R.; Pickard, C. J.; Zhu, Q.; Needs, R. J. Structure prediction drives materials discovery. *Nature Reviews Materials* **2019**, *4*, 331–348, DOI: 10.1038/s41578-019-0101-8.

(11) Wang, H.; Zhang, Z. Supporting the cif file format of proteins in molecular dynamics simulations. *Journal of University of Science and Technology of China* **2024**, *54*, DOI: 10.52396/JUSTC-2023-0148.

(12) Sandhu, A.; Chini, M. 2D and 3D Halide Perovskite-Based Supercapacitors. *ChemistrySelect* **2024**, *9*, DOI: 10.1002/slct.202304441.

(13) Yi, Z.; Ladi, N. H.; Shai, X.; Li, H.; Shen, Y.; Wang, M. Will organic–inorganic hybrid halide lead perovskites be eliminated from optoelectronic applications? *Nanoscale Advances* **2019**, *1*, 1276–1289, DOI: 10.1039/C8NA00416A.

(14) Shao, M.; Bie, T.; Yang, L.; Gao, Y.; Jin, X.; He, F.; Zheng, N.; Yu, Y.; Zhang, X. Over 21% Efficiency Stable 2D Perovskite Solar Cells. *Advanced Materials* **2022**, *34*, DOI: 10.1002/adma.202107211.

(15) Osilla What Are 2D Perovskites?, https://www.ossila.com/pages/what-are-2d-perovskites.

(16) Grancini, G.; Nazeeruddin, M. K. Dimensional tailoring of hybrid perovskites for photovoltaics. *Nature Reviews Materials* **2019**, *4*, 4–22, DOI: 10.1038/s41578-018-0065-0.

(17) Yu, P. Y.; Cardona, M., *Fundamentals of Semiconductors*; Springer Berlin Heidelberg: Berlin, Heidelberg, 2010, DOI: 10.1007/978-3-642-00710-1.

(18) Fang, T.-T. In *Elements of Structures and Defects of Crystalline Materials*, Fang, T.-T., Ed.; Elsevier: 2018, pp 83–127, DOI: https://doi.org/10.1016/B978-0-12-814268-4.00004-7.

(19) Kao, K. C. In *Dielectric Phenomena in Solids*, Kao, K. C., Ed.; Academic Press: San Diego, 2004, pp 115–212, DOI: https://doi.org/10.1016/B978-012396561-5/50013-X.

(20) Gatea, M. A.; Jumaah, G. F.; Al Anbari, R. H.; Alsalhy, Q. F. Review on Decontamination Manners of Radioactive Liquids. *Water, Air, & Soil Pollution* **2023**, *234*, 652, DOI: 10.1007/s11270-023-06678-x.

(21) Fouad, M. M.; Shihata, L. A.; Morgan, E. I. An integrated review of factors influencing the perfomance of photovoltaic panels. *Renewable and Sustainable Energy Reviews* **2017**, *80*, 1499–1511, DOI: https://doi.org/10.1016/j.rser.2017.05.141.

(22) Nande, A.; Raut, S.; Tanguturi, R. G.; Dhoble, S. J. In *Quantum Dots*, Thejo Kalyani, N., Dhoble, S. J., Michalska-Domańska, M., Vengadaesvaran, B., Nagabhushana, H., Arof, A. K., Eds.; Woodhead Publishing Series in Electronic and Optical Materials; Woodhead Publishing: 2023, pp 189–214, DOI: https://doi.org/10.1016/B978-0-323-85278-4.00019-2.

(23) Wu, G.; Liang, R.; Zhang, Z.; Ge, M.; Xing, G.; Sun, G. 2D Hybrid Halide Perovskites: Structure, Properties, and Applications in Solar Cells. *Small* **2021**, *17*, DOI: 10.1002/smll.202103514.

(24) Marchenko, E. I.; Fateev, S. A.; Petrov, A. A.; Korolev, V. V.; Mitrofanov, A.; Petrov, A. V.; Goodilin, E. A.; Tarasov, A. B. Database of Two-Dimensional Hybrid Perovskite Materials: Open-Access Collection of Crystal Structures, Band Gaps, and Atomic Partial Charges Predicted by Machine Learning. *Chemistry of Materials* **2020**, *32*, 7383–7388, DOI: 10.1021/acs.chemmater.0c02290.

(25) Brooks, B. R. et al. CHARMM: the biomolecular simulation program. *Journal of Computational Chemistry* **2009**, *30*, 1545–1614, DOI: 10.1002/jcc.21287.

(26) Sakai, Y.; Sen, S.; Sugihara, T.; Kakeyama, Y.; Iwasaki, M.; Schertler, G. F. X.; Deupi, X.; Koyanagi, M.; Terakita, A. Coral anthozoan-specific opsins employ a novel chloride counterion for spectral tuning. *eLife* **2025**, *14*, DOI: 10.7554/eLife.105451.

(27) Case, D. A. et al. AmberTools. *Journal of Chemical Information and Modeling* **2023**, *63*, 6183–6191, DOI: 10.1021/acs.jcim.3c01153.

(28) Case, D. A. et al. Recent Developments in Amber Biomolecular Simulations. *Journal of Chemical Information and Modeling* **2025**, *65*, 7835–7843, DOI: 10.1021/acs.jcim.5c01063.

(29) Sastre, G. In *Modelling and Simulation in the Science of Micro- and Meso-Porous Materials*, Catlow, C. R. A., Van Speybroeck, V., van Santen, R. A., Eds.; Elsevier: 2018, pp 27–62, DOI: https://doi.org/10.1016/B978-0-12-805057-6.00002-8.

(30) Mobarak, M. H.; Mimona, M. A.; Islam, M. A.; Hossain, N.; Zohura, F. T.; Imtiaz, I.; Rimon, M. I. H. Scope of machine learning in materials research—A review. *Applied Surface Science Advances* **2023**, *18*, 100523, DOI: https://doi.org/10.1016/j.apsadv.2023.100523.

(31) Zeng, Z. In *Proceedings of the 2nd International Conference on Data Science and Engineering - Volume 1: ICDSE*, SciTePress: 2025, pp 509–515, DOI: 10.5220/0013700300004670.

(32) Aziz, S.; Nayem, H. M.; Kibria, B. M. G. In *Proceedings of the 2025 Joint Statistical Meeting*, Zenodo: Nashville, USA, 2025, DOI: 10.5281/zenodo.17230601.

(33) Umoh, U. A.; Eyoh, I. J.; Murugesan, V. S.; Nyoho, E. E. In *Artificial Intelligence and Machine Learning for EDGE Computing*, Pandey, R., Khatri, S. K., kumar Singh, N., Verma, P., Eds.; Academic Press: 2022, pp 207–233, DOI: https://doi.org/10.1016/B978-0-12-824054-0.00025-3.

(34) Fridriksson, M. B.; van der Meer, N.; de Haas, J.; Grozema, F. C. Tuning the Structural Rigidity of Two-Dimensional Ruddlesden–Popper Perovskites through the Organic Cation. *The Journal of Physical Chemistry C* **2020**, *124*, 28201–28209, DOI: 10.1021/acs.jpcc.0c08893.

(35) Seijas-Bellido, J. A.; Samanta, B.; Valadez-Villalobos, K.; Gallardo, J. J.; Navas, J.; Balestra, S. R. G.; Madero Castro, R. M.; Vicent-Luna, J. M.; Tao, S.; Toroker, M. C.; Anta, J. A. Transferable Classical Force Field for Pure and Mixed Metal Halide Perovskites Parameterized from First-Principles. *Journal of Chemical Information and Modeling* **2022**, *62*, 6423–6435, DOI: 10.1021/acs.jcim.1c01506.

(36) Groom, C. R.; Bruno, I. J.; Lightfoot, M. P.; Ward, S. C. The Cambridge Structural Database. *Acta Crystallographica Section B* **2016**, *72*, 171–179, DOI: 10.1107/S2052520616003954.

(37) Abusaleh AA Wikibooks - Periodic Boundary Conditions, 2019.

(38) O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An open chemical toolbox. *Journal of Cheminformatics* **2011**, *3*, 33, DOI: 10.1186/1758-2946-3-33.

(39) Shirts, M. R.; Klein, C.; Swails, J. M.; Yin, J.; Gilson, M. K.; Mobley, D. L.; Case, D. A.; Zhong, E. D. Lessons learned from comparing molecular dynamics engines on the SAMPL5 dataset. *Journal of Computer-Aided Molecular Design* **2017**, *31*, 147–161, DOI: 10.1007/s10822-016-9977-1.

(40) Wu, J. In *Pantograph and Contact Line System*, Wu, J., Ed.; High-Speed Railway; Academic Press: 2018, pp 165–191, DOI: https://doi.org/10.1016/B978-0-12-812886-2.00005-7.

(41) Harris, C. R. et al. Array programming with NumPy. *Nature* **2020**, *585*, 357–362, DOI: 10.1038/s41586-020-2649-2.

(42) Image Upscaler ImgUpscaler: AI Image Upscaling and Enhancement, 2026.

(43) GitHub GitHub Copilot, https://github.com/features/copilot, 2024.

(44) OpenAI ChatGPT response to "how do i cite openais chatgpt", 2026.

Appendix

Use of Generative AI

This section describes the ways in which generative AI has been used throughout this research.

First, generative AI was used to upscale some pixelated images [42]. This did not require any prompts; after the upscaling the images were placed in the manuscript, which is also mentioned in the caption.

Furthermore, GitHub Copilot [43] was also used; no specific model was selected, it was left on automatic. Copilot was used for debugging code, especially in cases where the errors were not easily searchable or traceable. This was mostly the case during the creation of the pipeline, which required the use some lesser known packages, meaning less information or documentation was available. It should be noted that output from Copilot was never copied and always carefully checked.

Lastly, OpenAI's GPT-5.2 model [44] was used as an aid in writing the manuscript. For example, the model was asked for synonyms of certain words in an attempt to avoid repetition throughout the manuscript. Besides that, it was also used to grammar and spelling check the manuscript, especially towards finalising the manuscript. As with the Copilot model, GPT-5.2's outputs were not directly copied, and were always considered to be possibly incorrect.

It is important to note that generative AI outputs were always carefully checked and never assumed to be correct. The outputs were also not copied (with the exception of the upscaled images) or made to seem as if it is original work.