TEL AVIV UNIVERSITY
אוניברסיטת תל-אביב
THE IBY AND ALADAR FLEISCHMAN
FACULTY OF ENGINEERING
הפקולטה להנדסה
על שם איבי ואלדר פליישמן

The Unique Structure Prototypes of All Materials

A thesis
submitted in partial fulfilment
of the requirements for the Degree
of
Master of Science

by

Roee Asher

Tel Aviv University

December 2025

Abstract

Crystal structure offers a compact, composition independent basis for understanding materials behavior and for accelerating materials discovery. This thesis presents an automated and reproducible workflow for identifying structure prototypes directly from atomic geometries, rather than from database metadata. The study analyzes 23,160 experimentally observed thermodynamic ground-state inorganic materials from the Materials Project and compares structures using pymatgen's StructureMatcher with the FrameworkComparator, such that matches are determined by geometric frameworks irrespective of chemical species. A tolerance sensitivity analysis is used to set matching thresholds (length tolerance of 0.2, angle tolerance of 5°, site tolerance of 0.3), balancing robustness against over-matching while keeping computations tractable. Applying a prototype identification algorithm to the full ground-state set yields 6,898 unique frameworks, of which 2,410 are shared by two or more materials and were identified as prototypes. After additional filtering and stoichiometry guided partitioning, the workflow identifies 2,073 unique frameworks and 697 prototypes. The results show that the inorganic structural landscape is highly concentrated, with a small number of prototypes accounting for a disproportionately large fraction of known stable materials, and that nature favors higher symmetry systems due to the greater geometric variability of low-symmetry lattices under strict matching.

Table of Contents

List of Figures

List of Tables

1 Introduction

Atomic structure is the single greatest determinant of materials properties [1–3]. This can be evidenced by the differing properties between materials composed of the same elements but with different atomic arrangements. For elements, this phenomenon is called *allotropy* and for compounds or molecules, it is called *polymorphism*, but the underlying principle is the same. Allotropy can be typified by carbon in the differing forms of diamond and graphite. In diamond, each carbon atom is covalently bonded to four other carbon atoms in a three-dimensional tetrahedral arrangement. These strong covalent bonds create a rigid, continuous network throughout the entire structure, giving diamond remarkable hardness [4, 5]. In contrast, in graphite each carbon atom is covalently bonded to three other carbon atoms, forming flat, hexagonal layers. Within each layer, the bonds are strong, but the layers themselves are held together by much weaker forces, allowing them to slide past each other easily. As a result, graphite is soft and slippery, making it useful as a lubricant [6, 7]. Besides exhibiting different mechanical properties, they also differ in optical and electrical behavior: diamond is transparent and white [5, 8], whereas graphite is opaque and black [9, 10] (hence its common use in pencils). Furthermore, one is an insulator and the other capable of forming a superconductor [11–15]. Thus, atomic structure can be seen to greatly affect material properties on the macroscopic level even though graphite and diamond are both composed of only carbon.

While different structures of identical atoms lead to contrasting properties, the inverse is also true: materials composed of different elements but sharing the same crystal structure often exhibit broadly similar physical behavior. Such materials are commonly described as *isomorphous* or *isostructural* with respect to one another. For example, elemental silicon crystallizes in the same structure as diamond, forming a three-dimensional tetrahedral network that gives rise to qualitatively similar characteristics, such as high stiffness and brittle fracture along similar facets [16]. As is the case for carbon, forms of silicon with differing atomic bond topology do not display the diamond-like mechanical or electronic properties [17].

This phenomenon extends to compounds containing multiple elements. Diamond and zinc-blende (ZnS) both consist of a three-dimensional network of corner-sharing tetrahedra (see Figure 1.1(a), (b)), but while diamond contains a single atomic species, zinc-blende has zinc and sulfur occupying the same site arrangement in an ordered pattern [16]. As a result, ZnS has properties that are similar to diamond. The same can also be said of gallium arsenide (GaAs) and

indium phosphide (InP) *etc.* [18]. This shared atomic geometry leads to similar combinations of mechanical stiffness, electrical conduction, and other properties arising from the spatial arrangements of sites and bonds in this configuration, so both GaAs and InP are widely used in similar roles in the industry [19].

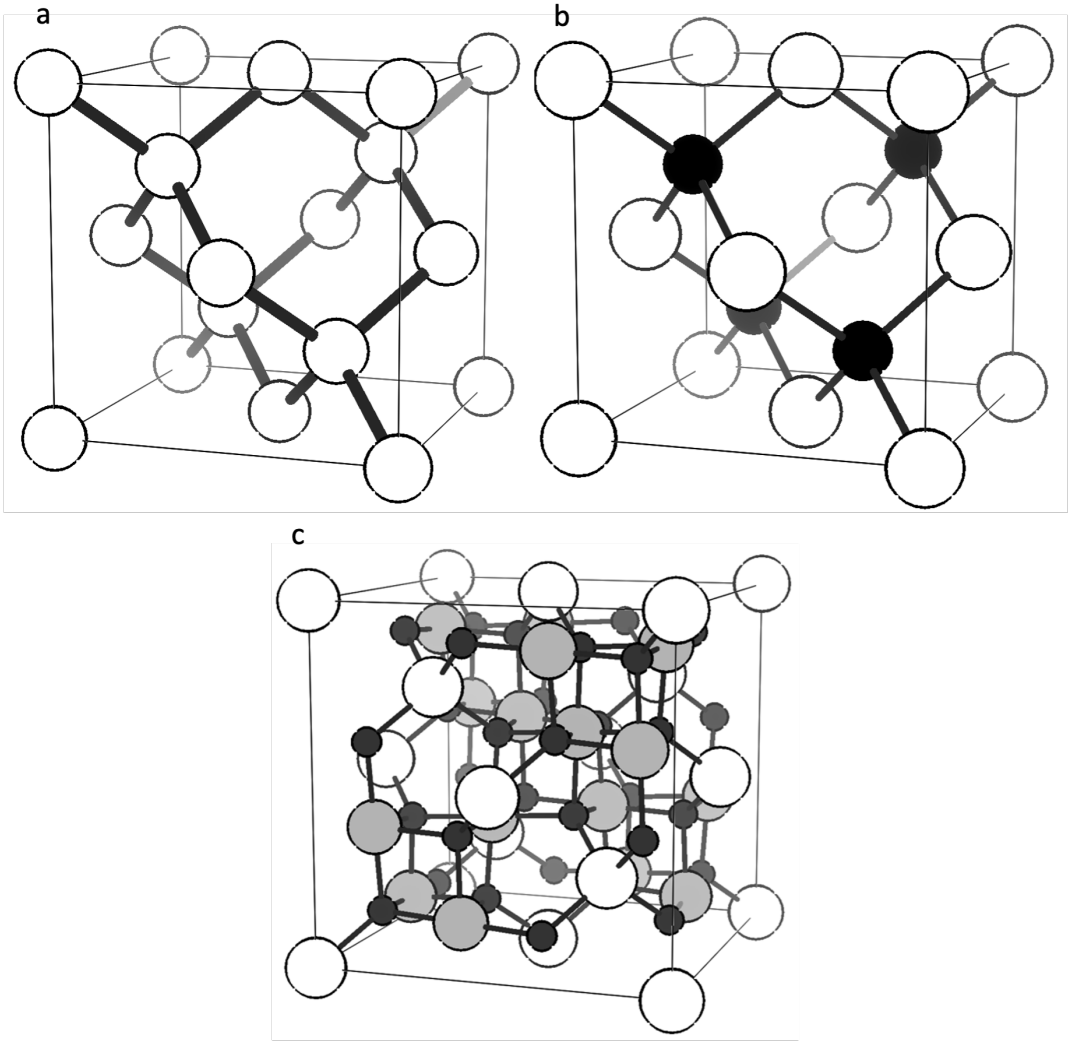


Figure 1.1: Unit cell structure of (a) diamond, (b) zinc-blende, and (c) spinel. The different colors signify different kinds of atoms.

To visualize the continuous three-dimensional networks characteristic of these structures, Figure 1.1 depicts them as bounded cubes. These cubes represent the fundamental repeating motif of the crystal, known as the "unit cell." A unit cell is defined as a specific volume of space containing atoms that, when translated in every spatial direction, reproduces the periodicity of the crystal structure. By reducing the extended atomic network to this finite description, it becomes possible to rigorously analyze and compare the precise geometric relationships between atoms.

The importance of structure over composition in determining macroscopic properties naturally led to different materials being categorized by shared atomic structure. For example, the aforementioned diamond and zinc-blende structures, as well as other structures such as rock-salt (named after NaCl) or perovskite ($CaTiO_3$), are also shared by materials with different chemical compositions. The consideration of structure, without regard to composition, is often called an atomic *framework*, focusing solely on the geometric arrangement of atomic positions. In this way, macroscopic properties of a material could be inferred from its mineral category, *e.g.*, diamond or perovskite, if one knew the underlying framework. Figure 1.1 shows how using this definition, both diamond and zinc-blende have the same framework, as they possess identical atomic connectivity and topology despite their compositional differences. However, labeling frameworks by mineral names can be confusing because it conflates a general geometry with a specific chemical compound. For instance, the term "zinc-blende" suggests the presence of zinc, yet it is routinely applied to compounds that contain no zinc at all. In addition, this labeling system requires knowledge of the framework already in order to be useful. For example, calling a structure zinc-blende-type does not tell you anything unless you already know what zinc-blende looks like. furthermore, some structures have multiple names. For example, zinc-blende is also referred to by the name "sphalerite", and rock-salt is called "halite". As the catalog of known structures increased, this mineral-based terminology became too inefficient and ambiguous for the task of materials classification, motivating the development of more standardized methods [20, 21].

The consideration of symmetry was a natural addition to the discourse surrounding materials structure. Symmetry describes an object that remains indistinguishable from its initial state after a certain transformation. Different atomic arrangements contain different combinations of symmetry elements. Thus, it might seem like a good way to distinguish between different materials structures. However, relying only on symmetry falls short of providing a definitive label for unique frameworks. For example, in the zinc-blende structure, the atomic sites are shared between two elements in an ordered fashion, which reduces the symmetry of the crystal relative to elemental diamond. However, introducing additional elements does not always lower the symmetry. For example, spinel ($MgAl_2O_4$) contains more elements than diamond, yet its atoms occupy additional sites that are empty in the diamond structure in a way that preserves the overall symmetry (Figure 1.1 (c)). These examples show that symmetry alone cannot fully determine whether two materials share the same framework.

The first systematic approach was the "Strukturbericht designation", proposed by Paul Peter Ewald in the early 20th century [22]. Each Strukturbericht designation is identified by a letter and a number, where the letter represents a specific stoichiometric composition or a group of similar compositions. For letters containing multiple compositions, number ranges are assigned for specific cases, although no strict rules for numbering were established. The letter "A" corresponds to single-element compounds, "B" to compounds containing two elements in equal proportions such as NaCl or ZnS, and "L" to intermetallic compounds. For example, A1 denotes the face-centered cubic (FCC) structure, A2 the body-centered cubic (BCC) structure, and A3 the hexagonal close-packed (HCP) structure. B1 corresponds to rock-salt (NaCl), B2 to CsCl, and B3 to zinc-blende

(ZnS). Although this system provided a pioneering system for structural identification, the generation of new symbols was eventually abandoned following World War II. As the number of newly determined structures grew, the symbolic nature of the assignments became unwieldy to extend. Consequently, when the International Union of Crystallography (IUCr) assumed responsibility for cataloging structural reports, the practice of assigning new Strukturbericht symbols was dropped [23]. However, the system is not purely of historical interest, as many of the original designations remain in widespread use throughout the scientific community today [23].

The limitations of the Strukturbericht system motivated the development of a more descriptive and logically structured notation. In the 1950s, William Burton Pearson introduced the "Pearson symbol" [24, 25], which encodes essential crystallographic information directly onto its label. A Pearson symbol consists of two letters and a number: the first letter (lowercase) denotes the crystal system, the second letter (uppercase) indicates the lattice centering, and the number specifies the number of atoms in the conventional unit cell. For example, the rock-salt structure has a cubic crystal system with face centering and eight atoms in the conventional unit cell, and is therefore denoted as cF8. This notation represented a clear improvement over the Strukturbericht designation by establishing a direct, logical connection between the label and the geometry of the structure. However, it also introduced new challenges: distinctly different structures can share the same Pearson symbol, since the notation does not account for atomic arrangement. For example, both diamond and rock-salt possess the same Pearson symbol, cF8, despite their fundamentally different bonding and atomic configurations.

With the rise of computational materials science, classification efforts have transitioned from manual to automated curation. The "Automatic FLOW for Materials Discovery" (`AFLOW`) platform was developed to standardize first-principles calculations, data analysis, and structure classification based on symmetry, atomic positions, and stoichiometry [21, 26]. Each structure is represented by a standardized label which includes the stoichiometry of the crystal, the Pearson symbol, the symmetry, and the atomic positions. For example, rock-salt is labeled as `AB_cF8_225_a_b-001`. This system improves upon earlier schemes by introducing a machine-readable and reproducible format suitable for large-scale data analysis. However, despite its automation, `AFLOW` still relies partly on manual validation, resulting in uneven structure coverage and potential biases toward well-known structure families. Furthermore, the dependence on discrete symmetry descriptors can obscure subtle distortions, polymorphic variations, and chemically driven deviations from ideal symmetry. Although effective for database organization, the symbolic labeling system of `AFLOW` is not human-readable and cannot easily impart intuitive information about the structure.

Together, these classification systems trace the arc of historical progression of structure identification in crystallography, from manually curated symbolic schemes to automated, symmetry-based databases. While these frameworks are essential for categorizing known materials, their most powerful application perhaps lies in predicting new materials. To facilitate this, the concept of a *prototype* was developed: a structural template shared among multiple materials with distinct chemical compositions. By treating a framework as a prototype, researchers can postulate the existence of future materials based on known stoichiometries. For example, CaO could reasonably

be assumed to crystallize in either the rock-salt or zinc blende structure types, in fact, it crystallizes in the rock-salt structure [27]. Furthermore, a newly predicted material adopting the zinc blende prototype could be inferred to possess specific physical properties, such as high hardness and rigidity.

Many technological breakthroughs are not the result of fundamental scientific discoveries, but from the utilization of superior materials [28]. A prime example is the jet engine, where thermodynamic efficiency is governed by the temperature difference between the hot combustion gases and the cooler surrounding air [29]. Since the maximum operating temperature is constrained by the thermal endurance of the engine components, manufacturing turbine blades from alloys that withstand higher temperatures allows greater efficiency [30]. This enhancement directly reduces fuel consumption, resulting in significant cost savings and reduction in pollution [31].

The ability to predict and design materials with such targeted physical properties is a critical driver of modern engineering. Indeed, many technological breakthroughs are not the result of fundamental scientific discoveries, but rather from the utilization of these superior materials [28]. A prime example is the jet engine, where thermodynamic efficiency is governed by the temperature difference between the hot combustion gases and the cooler surrounding air. Since the maximum operating temperature is constrained by the thermal endurance of the engine components, manufacturing turbine blades from advanced alloys that withstand higher temperatures allows for greater efficiency [30]. This enhancement directly reduces fuel consumption, resulting in significant cost savings and a reduction in pollution [31]. Consequently, the continuous demand for such high-performance applications strongly motivates the ongoing search for novel material phases.

Traditionally, the discovery of new materials was a process of experimental trial and error, involving the mixing of precursors to see if a novel material was made successfully. Today, this process is significantly enhanced by computer simulations [32]. Fundamentally, theoretical methods allow researchers to calculate the thermodynamic stability of a hypothetical atomic arrangement, determining whether a specific compound is likely to exist before any physical synthesis is attempted [32, 33]. This predictive capability serves as a crucial filter, guiding efforts toward viable candidates [34]. Furthermore, when applied at scale, these computational methods can screen vast numbers of materials, effectively bypassing the immense time and cost associated with exhaustive experimental testing [35, 36].

More sophisticated approaches to predict new materials can benefit directly from established structure frameworks. Modern structure prediction tools, such as the *Ab Initio Random Structure Search (AIRSS)* and the *Universal Structure Predictor: Evolutionary Xtallography (USPEX)* packages, intelligently employ randomness and flexibility within a structural framework in order to propose likely crystal structures for a given chemical composition, using density functional theory (DFT) to evaluate the energy of candidate atomic arrangements [37, 38]. The DFT calculations drive an iterative process toward a structure that represents the preferential phase for a material before any laboratory synthesis is attempted. However, such structure prediction workflows can often be made more reliable and substantially faster, since a typical search may require

thousands of DFT calculations, by incorporating relevant known structures into the generative process. These structures, sometimes referred to as "seed" structures, are structure prototypes that are agnostic with respect to element occupation. Providing a systematic, composition-independent catalog of such frameworks is therefore a natural way to support and enhance modern structure prediction methods.

As the scale of these computational searches expands, a consistent description of experimentally verified structure types becomes particularly important. A prominent recent example is Google DeepMind's *Graph Networks for Materials Exploration (GNoME)*, which reported predictions for about 2.2 million materials and 45,500 proposed new prototypes [39]. However, subsequent analyses indicate that a significant portion of these predicted structures are likely variations of known types rather than fundamentally new arrangements [40, 41]. These misclassifications often occur because automated algorithms can mistake minor geometric distortions for distinct structural frameworks. In light of this ambiguity, evaluating AI-generated predictions requires a clear and systematically defined catalog of experimentally established structure types. Such a reference is essential to distinguish whether proposed candidates truly expand the structural landscape or merely reconfigure existing motifs.

This introduction has motivated the need for a systematic notion of structural equivalence that remains reliable as the catalog of known inorganic crystals continues to expand and as case-by-case comparison becomes impractical. The remainder of the thesis develops the methodological basis for this goal, applies it to a broad set of experimentally determined structures to construct and analyze consistent structure-type groupings, and discusses the implications for how structure types are reported, compared, and reused across the literature and databases. In doing so, the work provides a concrete route from classical crystallographic descriptions to scalable, reproducible structure classification that remains meaningful for contemporary data-driven research.

2 Methods

The Introduction chapter established the importance of crystal structure. In this chapter, the definitions involved in describing crystal structure are discussed in more details. In addition, the workflow used to assign structure prototypes is described.

2.1 Geometrical Description of Crystals

The definition of a crystal is a solid material with a repeating ordered structure of atoms, ions, or molecules[42]. A crystal structure consists of a unit cell (a repeating unit) and a lattice (which dictates how the unit cell is repeated) [43]. A lattice can be thought of as an infinite set of discrete points arranged periodically in all spatial directions [43]. In such a lattice, two observers located at different lattice points would see exactly the same local environment. Mathematically, a lattice is described by a set of linearly independent basis vectors in N-dimensional space, denoted as $\{\vec{a}_i\}_{i=1}^{N}$, where the curly brackets indicate a set and the arrows over the letters denote vectors. This condition of linear independence ensures that no single vector can be expressed as a linear combination of the others. Every lattice point $\vec{R}$ can then be defined as an integer linear combination of these basis vectors:

$$\vec{R} = \sum_{i=1}^{N} n_i \vec{a}_i \tag{2.1}$$

Where $\vec{R}$ is a lattice point, n_i are integers, and $\vec{a}_i$ are the lattice vectors. While the lattice vectors do not have to be orthogonal, it is often convenient to represent them by an orientation matrix U_A. This matrix is constructed by projecting the lattice vectors on a cartesian coordinate system:

$$U_A = \begin{pmatrix} a_{1,x} & a_{2,x} & a_{3,x} \\ a_{1,y} & a_{2,y} & a_{3,y} \\ a_{1,z} & a_{2,z} & a_{3,z} \end{pmatrix} \tag{2.2}$$

Where $a_{i,x}, a_{i,y}$ and $a_{i,z}$ are the components of $\vec{a}_i$ in the x, y and z directions. This matrix allows for a unitless representation of atomic coordinates inside the unit cell as a fractional length of the lattice vectors. Furthermore, the orientation matrix directly encodes the macroscopic size of the crystal's repeating unit. Geometrically, the determinant of a matrix composed of three basis vectors

represents the volume of the parallelepiped spanned by those vectors. Therefore, the volume of the unit cell can be calculated by taking the absolute value of the determinant of the orientation matrix: $V = |det(U_A)|$.

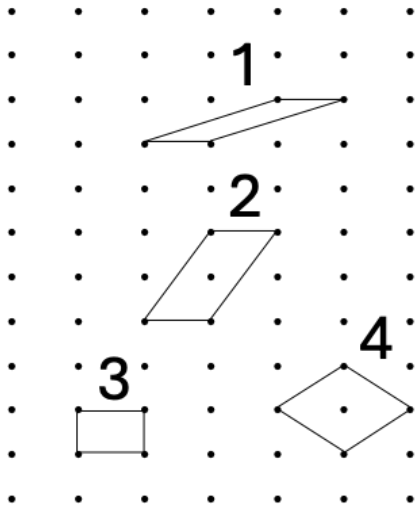


Figure 2.1: A 2-D lattice with various possible unit cells choices. The numbers 1-4 represent different unit cells that describe the same lattice.

A unit cell is the region of space spanned by a chosen set of lattice vectors. For every lattice, there is no unique choice of lattice vectors: there are infinitely many possible sets of lattice vectors, each defining a different unit cell. One property that is shared among every unit cell is that its shape tessellates space. That is, copying and pasting the unit cell along every direction of the chosen lattice vectors will fill the entire space. Figure 2.1 shows (a part of) a two dimensional lattice to illustrate this point.

A particular choice of lattice vectors defines the "lattice parameters", which are numerical quantities that describe the lattice. They consist of the magnitudes of the lattice vectors and the angles between them. In three dimensions, there are six lattice parameters (three lengths and three angles). The usual notation is to write the lengths of the lattice vectors (indicated by a vector between two vertical bars) $|\vec{a}_1|$, $|\vec{a}_2|$, and $|\vec{a}_3|$ as a, b, and c, respectively, and the angles between $\vec{a}_2$ and $\vec{a}_3$, between $\vec{a}_3$ and $\vec{a}_1$, and between $\vec{a}_1$ and $\vec{a}_2$ as α, β and γ, respectively [44].

A *primitive* unit cell is defined as a unit cell that contains exactly one lattice point. As illustrated in Figure 2.1 for the two dimensional case, while all the parallelograms shown qualify as unit cells, only 1 and 3 are primitive. However, the definition of a primitive cell alone does not uniquely constrain the choice of basis vectors. In contrast to primitive unit cells, a unit cell that contains more than one lattice point is referred to as a *non-primitive* unit cell. In Figure 2.1, unit cells 2 and 4 are non-primitive. A prominent class of non-primitive cells consists of the conventional centered unit cells, which contain additional lattice points at specific symmetry-defined

locations: at one pair of faces (base-centered), at the center of the unit cell (body-centered), or on all faces (face-centered). Another important category of non-primitive unit cells is the *supercell*, which is constructed by extending the lattice vectors of a "parent" unit cell to create a larger periodic domain with a volume that is an integer multiple of the original.

Different lattices are distinguished primarily by their symmetry. In three dimensions there exist seven distinct crystal systems, each with its own characteristic symmetry. For each system, the IUCr adopts conventional relations between the lattice parameters and a standard unit cell. Table 2.1 summarizes these seven 3-D crystal systems and their IUCr lattice-parameter conventions [44].

It follows from Equation 2.1 that the environment around every lattice point is identical; therefore, any choice of origin is equivalent. This property is called *translational symmetry*. In general, a symmetry operation is a transformation that leaves an object indistinguishable from its initial configuration. Two broad classes of symmetry are distinguished: point symmetries and space symmetries. Point symmetry leaves at least one point fixed, while space symmetry leaves no point in its original place. Typical point symmetry operations include rotation, inversion, and mirror reflection, and an example for a space symmetry operation is the lattice translation. Because lattices are distinguished by the set of symmetry elements they possess, symmetry provides the fundamental basis for classifying lattices and crystal structures [43]. In a crystallographic description, symmetry operations are often represented using the Seitz notation $\{\mathbf{S}|\vec{t}\}$, where $\mathbf{S}$ is an $N \times N$ rotation matrix and $\vec{t}$ is a translation vector. To illustrate this, the Seitz symbol and corresponding coordinate transformation for a 4_1 screw axis are presented below:

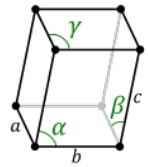
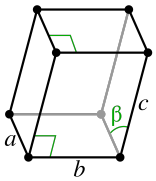
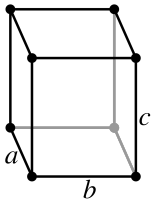
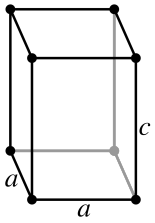
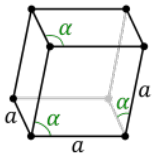
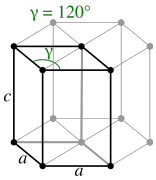
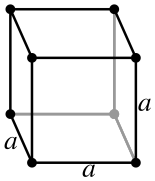
$$\{\mathbf{S}|\vec{t}\} = \left\{ \left(\begin{array}{ccc} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{array} \right) \middle| \left(\begin{array}{c} 0 \\ 0 \\ 1/4 \end{array} \right) \right\} \tag{2.3}$$

$$\mathbf{S} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \vec{t} = \begin{pmatrix} -y \\ x \\ z+1/4 \end{pmatrix} \tag{2.4}$$

This operation serves as an example of space symmetry, combining a rotation with a translation parallel to the rotation axis. The notation 4_1 specifies the exact nature of this motion: the main index (4) indicates a four-fold rotation ($360^\circ/4 = 90^\circ$), while the subscript (1) denotes a translation of $1/4$ of the lattice vector along the axis. In the Seitz notation above, the matrix $\mathbf{S}$ represents the rotation in the xy-plane transforming (x,y) into $(-y,x)$ while the vector $\vec{t}$ adds the translational component along z. The final result, shown in the second equation, maps an atom at position (x,y,z) to its symmetry-equivalent position at $(-y,x,z+1/4)$.

Applying these symmetry principles to the categorization of 3-D periodicity leads to the concept of the Bravais lattice. A specific combination of crystal system and centering constitutes a

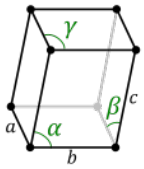
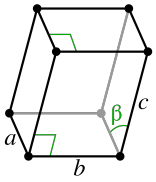
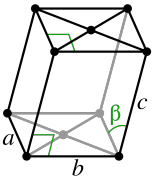
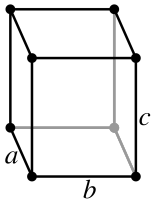
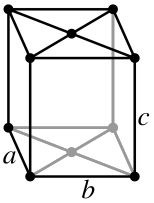
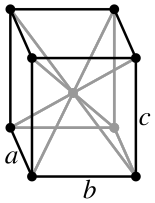
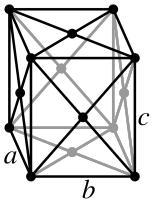
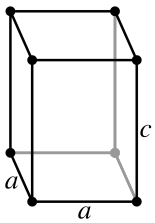
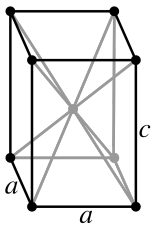
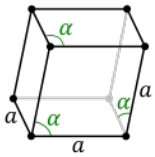
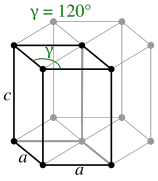
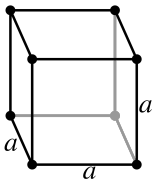
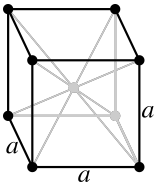
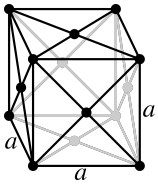
Table 2.1: The seven 3-D crystal systems. Images courtesy of Stannered via Wikimedia Commons (CC BY-SA 3.0).

Conventional Lattice Parameters	Crystal System	Unit Cell
$a \neq b \neq c$, $\alpha \neq \beta \neq \gamma$	Triclinic	
$a \neq b \neq c$, $\alpha = \gamma = 90°, \beta \neq 90°$	Monoclinic	
$a \neq b \neq c$, $\alpha = \beta = \gamma = 90°$	Orthorhombic	
$a = b \neq c$, $\alpha = \beta = \gamma = 90°$	Tetragonal	
$a = b = c$, $\alpha = \beta = \gamma$	Trigonal	
$a = b \neq c$, $\alpha = \beta = 90°, \gamma = 120°$	Hexagonal	
$a = b = c$, $\alpha = \beta = \gamma = 90°$	Cubic	

Bravais lattice, named after Auguste Bravais, who proved in 1850 that there are 14 distinct types of 3-D lattices [45]. It is important to note that not every possible combination of crystal system

and centering produces a geometrically unique lattice. The full list of three-dimensional Bravais lattices is given in Table 2.2 [43].

Table 2.2: The 14 3-D Bravais lattices. Images courtesy of Stannered via Wikimedia Commons (CC BY-SA 3.0).

Crystal system	Primitive	Base-centered	Body-centered	Face-centered
Triclinic				
Monoclinic				
Orthorhombic				
Tetragonal				
Trigonal				
Hexagonal				
Cubic				

Although the translational symmetry of any crystal can be described by a Bravais lattice, different symmetry arrangements of atoms can share the same Bravais lattice type. A finer level of classification is therefore required: the space group. A space group specifies all symmetry operations of a periodic crystal structure, including both the translational and point symmetries [44]. In three dimensions there are 230 distinct space groups. In this work, space groups are given using both the IUCr and Hermann–Mauguin conventions. The IUCr notation assigns an integer between 1 and 230 to each space group, while the Hermann–Mauguin notation uses an alphanumeric symbol that begins with a capital letter indicating the centering of the unit cell (for example P, I, F, C, or R, indicating primitive, body-centered, face-centered, C-base centered and rhombohedral, respectively), followed by one to three characters that denote characteristic symmetry elements along specific directions [44]. The total number of symmetry operations in a given space group is called the order of the group. For example, the rock-salt structure belongs to the space group $Fm\bar{3}m$, No. 225; the order of this group is 192 [44]. However, as discussed earlier, inherently different structures can belong to the same space group (*e.g.* diamond and spinel). Thus, symmetry alone is not sufficient to assign structure prototypes.

To specify an actual crystal structure, it is also necessary to describe how atoms occupy symmetry-related sites within the unit cell. Inside the unit cell, sets of symmetrically equivalent points are classified as *Wyckoff positions*. These sites are named after the American crystallographer Ralph Walter Graystone Wyckoff, who introduced a systematic method for tabulating the coordinates of symmetry-equivalent points in 1922 [46]. This standardization established the foundation used today in the *International Tables for Crystallography* [44, 47, 48]. There are two types of Wyckoff positions: special and general positions. Special positions are those left unchanged by at least one symmetry operation; consequently, the number of such equivalent positions is lower than the order of the group. The general positions have a multiplicity equal to the order of the group. Each Wyckoff position is denoted by a number indicating its multiplicity and a letter, beginning with *a* for the positions with the fewest equivalents (highest symmetry) and continuing alphabetically with ascending multiplicity. Every Wyckoff position describes a set of symmetry equivalent atomic coordinates. While some Wyckoff positions are defined by fixed numerical coordinates (*e.g.* $(0,0,0)$), others possess degrees of freedom represented by variable parameters x, y, or z (fractional coordinates between 0 and 1). For a given crystal structure, the apparent set of occupied Wyckoff positions can change if a different origin or cell setting is chosen, even though the underlying symmetry and geometry remain the same. Table 2.3 shows an example of the different Wyckoff positions for the space group *Pnma* (62). The site symmetry column in Table 2.3 utilizes standard Hermann-Mauguin point group notation to describe the local symmetry environment of each Wyckoff position. These symbols indicate which point symmetry operations leave the specific atomic site unchanged. The symbol 1, corresponding to the 8d position, indicates a completely asymmetric site. The only symmetry operation that leaves this local environment unchanged is the identity operation, and because it possesses no special symmetry, it has the highest multiplicity in the unit cell. The symbol $\bar{1}$, associated with the 4a and 4b positions, denotes a center of inversion. An atom sitting at this site is located exactly on an inversion center, meaning

the local environment is perfectly symmetric under spatial inversion. Finally, the symbol *.m.* for the 4c position represents a mirror plane. The dots serve as directional placeholders corresponding to the primary crystallographic axes of the space group. In this specific case, the notation *.m.* indicates that the mirror plane is oriented perpendicular to the *y*-axis, while no symmetry elements are present along the *x* and *z* axes.

Table 2.3: Wyckoff positions of space group *Pnma* (62) [44].

Multiplicity	Letter	Site symmetry	Coordinates
8	d	1	(x,y,z), $(-x+\frac{1}{2},-y,z+\frac{1}{2})$, $(-x,y+\frac{1}{2},-z)$, $(x+\frac{1}{2},-y+\frac{1}{2},-z+\frac{1}{2})$, $(-x,-y,-z)$, $(x+\frac{1}{2},y,-z+\frac{1}{2})$, $(x,-y+\frac{1}{2},z)$, $(-x+\frac{1}{2},y+\frac{1}{2},z+\frac{1}{2})$
4	c	*.m.*	$(x,\frac{1}{4},z)$, $(-x+\frac{1}{2},\frac{3}{4},z+\frac{1}{2})$, $(-x,\frac{3}{4},-z)$, $(x+\frac{1}{2},\frac{1}{4},-z+\frac{1}{2})$
4	b	$\bar{1}$	$(0,0,\frac{1}{2})$, $(0,\frac{1}{2},\frac{1}{2})$, $(\frac{1}{2},0,0)$, $(\frac{1}{2},\frac{1}{2},0)$
4	a	$\bar{1}$	$(0,0,0)$, $(0,\frac{1}{2},0)$, $(\frac{1}{2},0,\frac{1}{2})$, $(\frac{1}{2},\frac{1}{2},\frac{1}{2})$

The *Inorganic Crystal Structure Database (ICSD)* is a digital repository of experimentally determined inorganic crystal structures, enabling access and analysis by the global research community [49]. It contains over 300,000 entries, each representing a crystal structure accompanied by metadata such as chemical composition, oxidation states, space group, unit-cell parameters, and bibliographic references. The ICSD introduced the concept of *structure prototypes* to group structurally equivalent compounds [50]. According to the ICSD definition, two structures share the same prototype if they exhibit equivalent space groups, Wyckoff positions, Pearson symbols, identical numbers of atomic species, similar axial ratios (c/a), comparable β-angle ranges, and, in some cases, specific necessary or forbidden elements or atomic coordinates. This system provided a reproducible, symmetry-based standard for structural comparison. Nevertheless, it also introduced certain limitations, most notably its reliance on Wyckoff positions, which can vary depending on the choice of unit-cell setting or origin. As a result, geometrically identical structures may be assigned to different prototypes. Moreover, the use of Pearson symbols within this context is somewhat redundant, as they add little information once the equivalence of space group and Wyckoff positions is established.

In addition to structure types, the ICSD introduces the concept of a formula type [51]. This is a notation used to group materials by their stoichiometry. It labels cations (positively charged ions) with letters A-M, anions (negatively charged ions) are labeled with X, Y, Z, S-W, and neutral atoms labeled as N-R [51]. Letters are assigned alphabetically with ascending order of atoms present in the repeating unit (Thus, AB_2X_4 rather than A_2BX_4). Importantly, in case of multiplicities, the numbers are multiplied $Co(NO_3)_2$ will be noted as AX_2Y_6. Hydrogen is always ignored, even with available coordinates. This means molecular information can be lost and means the method

is impractical for non-inorganic periodic crystals.

Throughout this work, both concepts of prototypes and formula types are adopted. While formula types are used exactly as introduced by the ICSD, adjustments are made to the definition of structure prototypes. The ICSD classification relies on SQL queries and explicit data flags, grouping entries based on metadata such as space group numbers and Wyckoff sequences. This approach depends heavily on the accuracy of database labels. In contrast, this work prioritizes the physical structure over its description. Structures are compared relying solely on the actual geometrical arrangement of their atoms, independent of assigned tags like space group or Wyckoff positions. Therefore, a prototype is defined here as a unique geometric framework shared by multiple materials, identified through direct structural matching rather than metadata filtering.

Implementing this direct, geometry-based classification across thousands of materials presents a significant computational challenge. To identify these shared geometric frameworks, an algorithm must be capable of mathematically mapping lattice vectors and atomic coordinates to verify topological equivalence, completely ignoring the underlying chemical species. To determine the most appropriate computational tool for this exact task, an evaluation of several prominent structural matching algorithms is required.

The landscape of available structural comparison software includes packages such as StructureMatcher [52], AFLOW-XtalFinder [53], XTALCOMP [54], STRUCTURE-TIDY [55], CRYCOM [56], CMPZ [57], SPAP [58], and COMPSTRU [59–63]. The selection of an optimal algorithm for this research depends on three primary factors: programmatic accessibility, the capacity for high-throughput execution, and the ability to perform species-agnostic framework comparisons. While the availability of open-source code is not a strict prerequisite, it is highly important for conducting transparent, reproducible, and automated scientific research. Algorithms such as CMPZ, CRYCOM, STRUCTURE-TIDY, and COMPSTRU lack publicly available source code. This restricted access makes it difficult to seamlessly integrate them into customized data pipelines. Furthermore, COMPSTRU requires *a priori* symmetry input and operates primarily as an online tool hosted on the Bilbao Crystallographic Server, preventing autonomous execution on local computational clusters. Among the open-source options, the capacity for high-throughput evaluation is essential due to the scale of known materials. Tools like XTALCOMP and SPAP primarily focus on single pairwise comparisons, lacking the built-in architecture to efficiently perform multiple comparisons across thousands of structures without extensive external scripting. In contrast, StructureMatcher and AFLOW-XtalFinder are explicitly designed to handle high-throughput computational environments.

Finally, while tools like AFLOW-XtalFinder also feature structure-type comparison modes, the decision to utilize StructureMatcher is driven by its programmatic flexibility and seamless integration into modern Python-based computational pipelines. Utilizing the `pymatgen` library allows for the direct parsing and structural manipulation of crystallographic data as in-memory objects. This eliminates the risk of data loss, precision truncation, or parsing errors frequently associated with writing and reading intermediate file formats for standalone executable programs.

Furthermore, the core methodology of this research relies on an extensive tolerance sensitivity analysis. The StructureMatcher application programming interface (API) provides direct, dynamic control over its continuous fractional parameters, such as length, angle, and site tolerances (to be discussed later). This architectural flexibility enables the high-throughput, iterative tuning required to establish the optimal geometric matching thresholds for prototype identification.

Having established the `pymatgen` library [52] as the optimal computational environment for large-scale structural comparison, the StructureMatcher module is deployed as the central computational engine of this work. It evaluates pairs of crystal structures and outputs a Boolean value indicating geometric alignment. Crucially, because the primary objective of this research is to analyze the underlying topology of crystals without regard to their specific chemical composition, the module is configured using the FrameworkComparator class. This specific comparator maps two structures strictly according to their occupied crystallographic sites, treating all atomic species identically. This functionality directly fulfills the geometry-driven definition of a structure prototype established for this study.

StructureMatcher has multiple arguments that could be useful for different cases. For the purpose of this work, the most relevant ones are the type of comparator used and the different tolerances for matching structures. The rest are related to creating a supercell for matching.[1] StructureMatcher employs a validation algorithm to determine structural equivalence. This process is designed to address the challenge of non-unique crystal representations, where a single physical structure can be described by infinite choices of basis vectors and unit cell origins.

1. The first step is creating a standardized unit cell for each material. Creation of a primitive unit cell is performed by identification of symmetry operations. This process is done by `spglib` [64].

    1.1. If the provided input unit cell is not primitive, it inherently contains multiple lattice points. `spglib` first attempts to locate these inner lattice points by searching for pure translation operations, denoted as $\{\mathbf{I}|\vec{t}\}$, where $\mathbf{I}$ is the identity matrix, that successfully map the crystal structure onto itself. To achieve this efficiently, candidate translation vectors, $\vec{t}^{\,c}_{\mathbf{I}}$, are generated by taking the difference in positional coordinates between a fixed atomic site and all other atomic sites belonging to the same chemical species. To minimize the computational cost of this brute-force step, the algorithm intelligently selects the fixed atom from the chemical species that has the smallest number of atoms in the unit cell.

    Each candidate translation is then tested. A translation is accepted if it maps every atom in the cell to another atom of the same species such that the Cartesian distance between the mapped position and the target position is less than the tolerance parameter ε. If the input cell is already a primitive cell, only the identity translation

[1] The full list of arguments is available in the `pymatgen` documentation: https://pymatgen.org/pymatgen.html

$\vec{t}^{c}_{\mathbf{I}} = (0,0,0)^{\top}$ will be found. Otherwise, a set of multiple pure translation vectors is returned.

1.2. Once the set of pure translations is obtained, the algorithm constructs the basis vectors for the primitive cell $(\vec{a}_{1,p}, \vec{a}_{2,p}, \vec{a}_{3,p})$. The candidates for these new basis vectors are chosen from a combined pool containing the original basis vectors of the input cell and the newly discovered pure translation vectors.

The selection is constrained by two primary geometric criteria:

i. The chosen vectors must form a right-handed coordinate system.

ii. The volume of the resulting primitive cell, V_p, must satisfy the expected ratio constraint:

$$|T_p| = \left(\frac{V_i}{V_p}\right) \tag{2.5}$$

where V_i is the volume of the initial input cell, and $|T_p|$ is the total number of pure translations found (including the identity translation).

Structural distortions may cause the algorithm to find either too many or too few pure translations, causing the volume constraint to fail. To resolve this, `spglib` utilizes an iterative fallback mechanism:

- Thinning out translations: If an inconsistency is detected, `spglib` tightens (reduces) the tolerance parameter ε and re-examines the previously found pure translations. Translations that no longer satisfy the tightened tolerance are discarded. This "thinning" process is optimized, as it avoids restarting the computationally expensive global translation search from scratch.
- Restarting the search: If iterating through the thinning process fails to yield a valid primitive basis, the algorithm safely aborts the current loop and restarts the entire pure translation search using the newly reduced tolerance value.

Once a valid set of primitive basis vectors satisfying all crystallographic constraints is established, the atoms are mapped into the new coordinate system to finalize the creation of the primitive cell. This newly standardized primitive cell then serves as the foundation for the subsequent stages of the `spglib` algorithm, specifically the extraction of the point group operations and the identification of the exact space-group type.

2. Once `spglib` successfully identifies a primitive unit cell, the structure is stripped of unnecessary translational redundancies. However, because a single crystal lattice can be described by an infinite number of primitive basis sets, comparing two arbitrary primitive cells directly is prone to error. To address this ambiguity, this work utilizes the *Niggli-reduced* cell. This concept is named after Paul Niggli, who proved in 1928 that a crystal lattice can be characterized by a unique choice of a reduced cell [65]. The Niggli-reduced cell is defined by the shortest possible basis vectors that satisfy a set of mathematical conditions [66, 67].

This standardization ensures that any given lattice yields a unique and reproducible set of parameters, regardless of the initial unit cell choice.

The Niggli reduction process uses six scalar quantities, which are defined as:

$$\begin{aligned} A &= \vec{a}_1 \cdot \vec{a}_1 = a^2 \\ B &= \vec{a}_2 \cdot \vec{a}_2 = b^2 \\ C &= \vec{a}_3 \cdot \vec{a}_3 = c^2 \\ \xi &= 2\vec{a}_2 \cdot \vec{a}_3 = 2bc\cos\alpha \\ \eta &= 2\vec{a}_1 \cdot \vec{a}_3 = 2ac\cos\beta \\ \zeta &= 2\vec{a}_1 \cdot \vec{a}_2 = 2ab\cos\gamma \end{aligned}$$

Using these quantities, a set of conditions is checked. The main conditions are:

2.1. The basis vectors are ordered by magnitude: $A \leq B \leq C$, which implies $a \leq b \leq c$. If this condition is not met, the lattice vectors are swapped in order to satisfy the condition. In case of equality, another condition is imposed: if $A = B$, then the condition is $|\xi| \geq |\eta|$, and if $B = C$, then the condition is $|\eta| \geq |\zeta|$.

2.2. All angles are either all acute or all obtuse:

- Type I: $\xi > 0, \eta > 0, \zeta > 0$
- Type II: $\xi < 0, \eta < 0, \zeta < 0$

mixed signs are forbidden (*e.g.*, $\xi > 0, \eta < 0$). If there are mixed signs, than a transformation of lattice vectors of the type $\vec{a}'_i = -\vec{a}_i$ is performed.

2.3. The projection of any axis onto another must be less than or equal to half the length of that axis. This ensures the vectors are as orthogonal as possible.

$$\begin{aligned} |\xi| &\leq B \\ |\eta| &\leq A \\ |\zeta| &\leq A \end{aligned}$$

If $|\xi| > B$, then the applied transformation is $\vec{a}'_3 = \vec{a}_3 - \text{sign}(\xi)\vec{a}_2$. Similarly, for the other inequalities, the transformations are $\vec{a}'_3 = \vec{a}_3 - \text{sign}(\eta)\vec{a}_1$ and $\vec{a}'_2 = \vec{a}_2 - \text{sign}(\zeta)\vec{a}_1$, respectively.

In case of equality, another condition is set to ensure uniqueness. For example, $|\xi| = B$ implies $\vec{a}_3 = \vec{a}_3 - \vec{a}_2$. in that case, the condition is $\zeta \leq 2\eta$. If this condition is not met, the same transformation of $\vec{a}'_3 = \vec{a}_3 - \text{sign}(\xi)\vec{a}_2$ is applied. If $|\eta| = A$, then ζ $le2\xi$, and if $|\zeta| = A$ then $\eta \leq 2\xi$.

After applying any of these transformations, the previous conditions must be met with the transformed lattice vectors.

2.4. The sum of the scalar products and squared lengths must be non-negative to ensure the body diagonal is not shorter than the axes:

$$A + B + \xi + \eta + \zeta \geq 0$$

This condition should only be checked for cells of type II, since in cells of type I all quantities are positive. In case of equality, the additional condition is $2A + 2\eta + \zeta \leq 0$. If this condition is not met, the transformation $\vec{a}'_3 = \vec{a}_1 + \vec{a}_2 + \vec{a}_3$ is applied. after applying this transformation, the previous conditions must be met.

3. With both structures reduced to their unique, Niggli-reduced primitive cells, the structural comparison algorithm can proceed. To optimize computational efficiency, the matcher does not immediately evaluate individual atomic positions. Instead, it performs a preliminary filter by comparing the macroscopic geometric parameters of the two unit cells. If the foundational lattices are fundamentally incompatible, the structures are immediately deemed non-matching, bypassing the more expensive site-matching algorithms.

   The lattice matching process evaluates the compatibility of the three basis vector lengths (a, b, c) and the three inter-axial angles (α, β, γ) between the two reduced cells. Two lattices are considered equivalent if they satisfy two specific tolerance thresholds:

   3.1. Fractional Length Tolerance ($ltol$): The relative difference between the corresponding basis vector lengths of the two lattices must fall within a defined fractional threshold. For each lattice vector $i \in \{a, b, c\}$, the condition evaluated is:

   $$\frac{|l_{1,i} - l_{2,i}|}{\max(l_{1,i}, l_{2,i})} \leq ltol \tag{2.6}$$

   where $l_{1,i}$ and $l_{2,i}$ are the lengths of the i-th basis vector for the first and second structure, respectively.

   3.2. Absolute Angle Tolerance ($atol$): The absolute difference between the corresponding inter-axial angles must be less than a specified degree threshold. For each angle $j \in \{\alpha, \beta, \gamma\}$, the algorithm requires:

   $$|\theta_{1,j} - \theta_{2,j}| \leq atol \tag{2.7}$$

   where $\theta_{1,j}$ and $\theta_{2,j}$ are the j-th inter-axial angles of the first and second structure.

4. Following a successful lattice match, the algorithm must verify if the atomic motifs of the two structures are equivalent. Because the structures are represented in fractional coordinates within periodic boundary conditions, direct Cartesian distance calculations are insufficient. Furthermore, the two structures may be arbitrarily translated relative to one another.

The site matching algorithm must therefore find a global translation vector and a 1:1 mapping of atoms that minimizes the interatomic distances.

Let s_1 and s_2 represent the two structures, containing fractional atomic coordinates $\vec{x}_{1,i}$ and $\vec{x}_{2,j}$, respectively, where i and j iterate over the number of atoms N. To test for equivalence, the algorithm applies a trial fractional translation vector, $\vec{t}$, to the coordinates of s_2. The fractional difference vector between a site in s_1 and a translated site in s_2 is given by:

$$\Delta\vec{x}_{ij}(\vec{t}) = (\vec{x}_{2,j} + \vec{t}) - \vec{x}_{1,i} \tag{2.8}$$

To enforce the periodicity of the crystal lattice, the algorithm applies the Minimum Image Convention. This ensures that the shortest path between two atoms is evaluated, even if that path crosses a unit cell boundary. The periodic fractional difference vector, $\Delta\vec{x}_{ij}^{PBC}$, is computed by shifting the coordinates into the domain $[-0.5, 0.5]$:

$$\Delta\vec{x}_{ij}^{PBC}(\vec{t}) = \Delta\vec{x}_{ij}(\vec{t}) - \lfloor \Delta\vec{x}_{ij}(\vec{t}) + 0.5 \rfloor \tag{2.9}$$

Because fractional distances do not correlate with physical spatial distances in non-cubic systems, $\Delta\vec{x}_{ij}^{PBC}$ must be transformed into Cartesian space. Let $\vec{r}$ denote Cartesian coordinates. This transformation is achieved by multiplying the periodic fractional difference vector by the lattice matrix U_A, whose rows consist of the Cartesian basis vectors of the matched lattice:

$$\Delta\vec{r}_{ij}(\vec{t}) = U_A \Delta\vec{x}_{ij}^{PBC}(\vec{t}) \tag{2.10}$$

The true Euclidean distance between the mapped sites, d_{ij}, is the norm of this Cartesian vector:

$$d_{ij}(\vec{t}) = \sqrt{\Delta\vec{r}_{ij}(\vec{t})^\top \Delta\vec{r}_{ij}(\vec{t})} \tag{2.11}$$

For the structures to be considered a topological framework match, there must exist at least one valid permutation of the atomic sites, irrespective of their chemical identity, and one translation vector $\vec{t}$ such that the maximum atomic displacement is constrained by the average free length per atom, λ. This normalization factor is defined as:

$$\lambda = \left(\frac{V}{N}\right)^{\frac{1}{3}} \tag{2.12}$$

with V representing the volume of the matched unit cell and N being the total number of atomic sites. The matching condition requires that the maximum normalized displacement is less than or equal to a dimensionless fractional site tolerance parameter, $stol$, evaluated as:

$$\frac{\max_i(d_{i,\pi(i)}(\vec{t}))}{\lambda} \leq stol \tag{2.13}$$

where $\pi(i)$ represents a valid bijective mapping (permutation) from the index i in s_1 to an atom in s_2. Physically, this bijective mapping is a transformation that assigns every atom in s_1 to exactly one unique atom in s_2, ensuring that no two atoms in s_1 are mapped to the same site and no atom in s_2 is left unmatched

In practical implementations utilizing a framework comparator, discovering the optimal permutation $\pi(i)$ is formulated as a linear sum assignment problem. Rather than brute-forcing all $N!$ possible mappings, which is computationally unfeasible for large structures, the algorithm constructs an $N \times N$ cost matrix, C. Each element C_{ij} represents the physical spatial divergence between a site i in s_1 and a site j in s_2 under a given translation $\vec{t}$. Using the Euclidean distances calculated previously, the cost matrix is defined as:

$$C_{ij}(\vec{t}) = d_{ij}(\vec{t}) \tag{2.14}$$

The objective of the assignment problem is to find a boolean assignment matrix, where each element is 1 if site i is mapped to site j, and 0 otherwise. To ensure a valid bijective mapping, exactly one element in each row and each column must equal 1. The optimal mapping $\pi(i)$ is the permutation that minimizes the total sum of the mapped distances:

$$\min_{\pi} \sum_{i=1}^{N} C_{i,\pi(i)}(\vec{t}) \tag{2.15}$$

To solve this minimization efficiently in polynomial time, structural comparators utilize the Hungarian algorithm [68]. The procedure operates iteratively on the cost matrix:

4.1. Row and Column Reduction: The algorithm subtracts the minimum value of each row from all elements in that row, followed by subtracting the minimum value of each column from all elements in that column. This normalizes the matrix and creates zeros, which represent potential optimal, zero-relative-cost assignments.

4.2. Zero Covering: The algorithm attempts to cover all zeros in the matrix using the minimum possible number of horizontal and vertical lines.

4.3. Iterative Refinement: If fewer than N lines are required to cover all zeros, the optimal assignment is not yet complete. The algorithm finds the smallest uncovered value, subtracts it from all uncovered elements, and adds it to elements at the intersection of lines. This introduces new zeros.

4.4. Optimal Assignment: Steps 2 and 3 are repeated until exactly N lines are required to cover all zeros. At this point, the positions of N independent zeros define the optimal bijection $\pi(i)$.

Once the Hungarian algorithm identifies the optimal global mapping $\pi(i)$ that minimizes the overall spatial divergence, the algorithm evaluates this specific mapping against the maximum site tolerance constraint. If the maximum individual atomic displacement within this

optimal assignment satisfies the condition of equation 2.13, the structures are deemed a topological match for the given translation $\vec{t}$.

In addition, `pymatgen` serves as the core infrastructure for the *Materials Project* [35], an online database providing properties of inorganic materials derived from high-throughput first-principles calculations. The foundation of the Materials Project relies heavily on experimental structures sourced from the ICSD, which is used as a basis of experimentally characterized materials for DFT calculations of structure and properties. To process this data, the `pymatgen` StructureMatcher algorithm was utilized to map and group similar crystal structures, thereby eliminating duplicate entries and establishing a unique mapping between experimental records and computed materials. Over time, the repository has expanded significantly, as shown in Figure 2.2, incorporating not only these experimental structures, but also purely theoretical predictions, such as those generated by the *GNoME* deep learning tool.

DFT calculations assign an energy for each atomic configuration, and the one with the lowest energy is called the thermodynamically stable structure. The thermodynamic stability of a structure changes under different temperature and pressure, and the stable structure at zero Kelvin and one atmosphere is known as the *ground state*, while the structures above the ground state are metastable [33, 69]. The thermodynamic ground state, in turn, determines the predominant form that any combination of elements will take. For carbon allotropy, graphite is the thermodynamic ground state, while diamond is not the thermodynamic ground state and requires high temperature and pressure to form [70]. As a result, graphite is abundant in nature, while diamond occurs only under specific geological conditions, reflecting the strong connection between the atomic structure and thermodynamic stability.

For the purposes of this work, a curated subset of the Materials Project was selected. From the vast number of entries, the dataset was restricted to experimentally observed materials, excluding purely theoretical predictions. Furthermore, only the ground state structures were retained, ensuring that every chemical formula is represented by a unique crystal structure. This filtering step was necessary to resolve ambiguities arising from materials that exhibit polymorphism or stacking variations; for example, layered materials like MoS_2 can exist in multiple nearly identical structural forms [71]. At the time of writing this thesis, the resulting dataset comprises 23,160 materials.

2.2 Materials Comparison & Prototype Identification

Using the ground state materials from the materials project and StructureMatcher, the prototype identifying process could take place. This iterative process goes over all the materials in the dataset and compares them with the materials that were previously checked, as can be seen in Figure 2.3. The full code can be seen in the appendix, at subsection A.1.1.

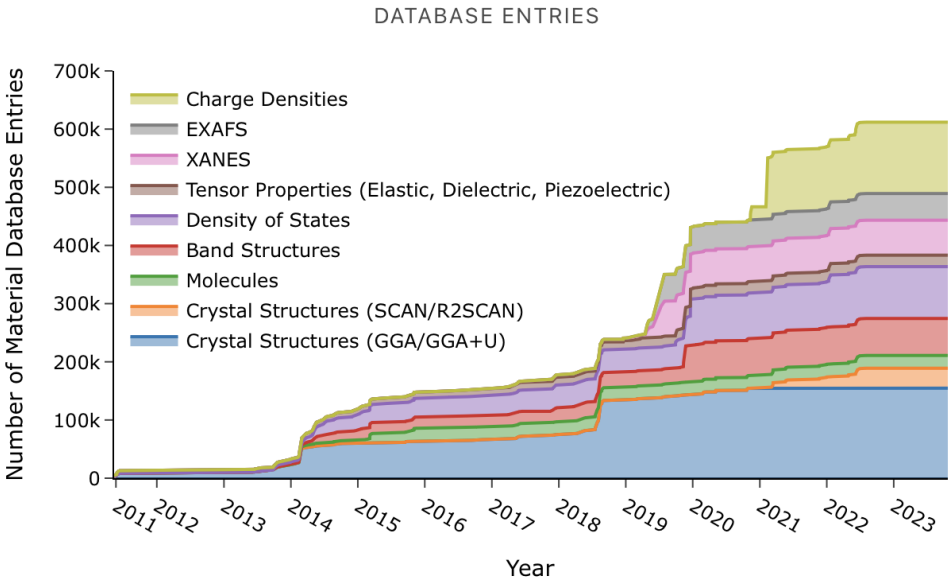


Figure 2.2: Growth of the Materials Project database from 2011 to 2023, showing the increasing number of material entries and diversity of attributes generated through different computational methods.

A pairwise comparison of n entries scales the number of matching operations as n^2: there are n options to choose the first material, which are multiplied by the $n-1$ options to choose a second one, then divided by two to avoid double-counting. Using a pairwise comparison for 23,160 materials would mean 2.68×10^8 operations. At this scale, if every matching operation took one second, the matching process would last about 8.5 years. However, in this work, a different approach is implemented. Under the assumption that if material A matches material B and material B matches material C, then material A also matches material C, the problem can become manageable. In this approach, each material is compared only against a list of representative structures. This not only reduces computational cost, but also groups the dataset into structural prototypes.

Complementing the global structural comparison, a refined matching protocol was implemented to restrict comparisons exclusively to materials sharing the same stoichiometry. To facilitate this partitioning, the dataset underwent additional pre-processing, which included the application of specific quality filters and the assignment of ANX notation to each compound, as described below.

1. The material must not contain hydrogen, as it usually does not have a fixed position in the unit cell, as well as being undetectable by X-ray diffraction (XRD) [72]. In addition, hydrogen does not have an assigned letter according to the ICSD formula type. Performing this step removed 1,457 materials from the original 23,160 in the dataset.

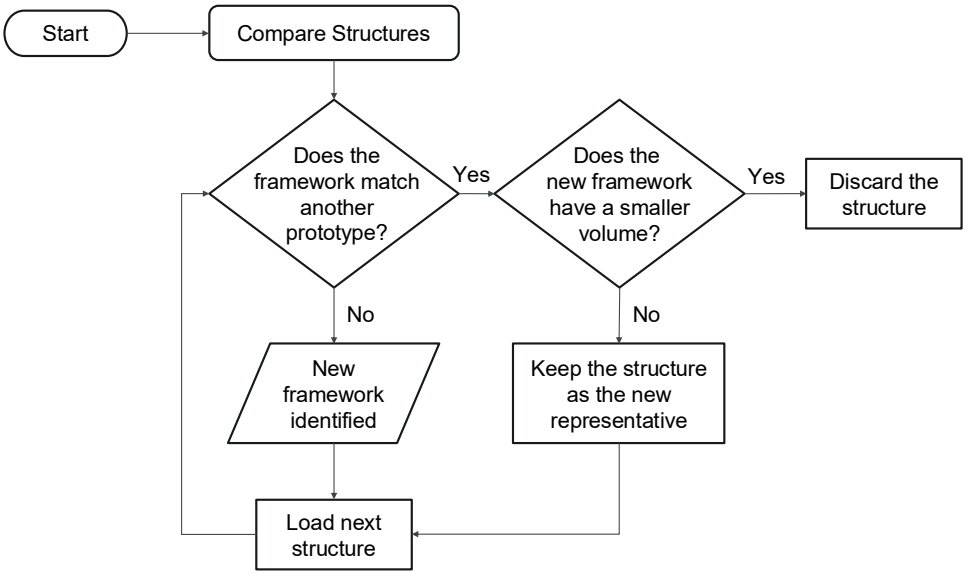


Figure 2.3: Flowchart of the prototype-identifying algorithm beginning in the top left.

2. The material must not contain noble gases. This filter was applied to mitigate errors that arise when processing these compounds, primarily because they have no assigned electronegativity. This step removed 64 materials.

3. Materials that contain more than 100 atoms in the unit cell were removed. This step removed 496 materials.

4. Materials that have more than 4 instances of the same atom in the repeating unit (*e.g.* AX_6), were removed. This step removed 9,190 materials.

5. Materials that have more than 3 different species of cations or anions were removed (*e.g.* $ABCDX_2$). this step removed 71 materials.

Filters 3-5 were done to remove complex materials and save computational runtime. After applying these filters, the dataset contained 5,557 materials.

A study by Ding *et al.* screening the ICSD found that most elements almost always exist in one form, either anion or cation [73]. As such, chemical formulae can be reliably reduced to their agnostic formula type and *vice versa*, allowing for even unknown chemical formula to be related to known frameworks. In this work, this premise is used to assign a formula type label to materials by chemical formula. The full code can be seen in the Appendix, at subsection A.1.2. An interesting case is the element arsenic, it is the only element that appears with both positive and negative oxidation states in the study by Ding et al. After careful considerations, including counting the

number of times it appeared with positive and negative oxidation states (7,207 positive, 3,143 negative), it was ultimately decided that arsenic would be regarded as a cation in our study, which otherwise is the only discrepancy in the oxidation states between our work and the previous study.

Materials that were found to not contain cations or anions were also removed from the dataset. this step was done to remove molecular crystals (*e.g.* N_2 or CO_2), and metallic compounds, as they tend to share similar prototypes across different stoichiometries. this step removed 6,325 materials.

Finally, the selection of the representative structure for each prototype involved a two-step logic. During the matching process, the algorithm was configured to always retain the structure with the largest unit cell volume as the temporary representative. This strategy was chosen to facilitate future structure relaxation: starting with a larger volume reduces the likelihood of atomic overlap and instability in initial DFT calculations. However, this technical preference often selected compounds with high atomic numbers and large ionic radii (*e.g.*, CsI for rock-salt), which are less intuitively recognizable than their common counterparts. To address this, the final representatives were cross-examined to identify the isostructural compound composed of the most common elements in the Earth's crust [74]. The abundance score, A, was calculated as the geometric mean of the elemental abundances:

$$A = \left(\prod_{i=1}^{n} x_i \right)^{1/n} \tag{2.16}$$

where x_i is the crustal abundance of the i-th element and n is the total number of atoms in the formula unit (*e.g.* 5 for Al_2O_3, 2 for NaCl, 7 for $MgAl_2O_4$, *etc.*). Applying this metric ensures that the prototypes analyzed in the following chapter are represented by their most chemically familiar examples, bridging the gap between computational utility and general readability.

With the matching criteria and representative selection process established, this methodology was applied to the materials dataset. The following chapter presents the results of this analysis, examining the distribution and characteristics of structure prototypes in both the unfiltered and filtered datasets.

3 Results & Discussion

In this chapter, the results output from the process described in the Methods chapter will be shown and discussed.

Prior to presenting the main results, the ground state materials dataset is analyzed with a specific focus on space group distribution. To gauge how well this dataset represents the broader landscape of inorganic materials, the findings are compared to the work of Baur and Kassner [75], who surveyed 34,692 inorganic structures from the ICSD. Analysis of the dataset used in this work reveals that certain space groups appear with significantly higher frequency than others. The most common space group in the dataset is *Pnma* (No. 62), comprising 2,321 materials (10.0% of the dataset). It is followed closely by $P2_1/c$ (No. 14) with 2,235 materials (9.7%) and *C2/c* (No. 15) with 995 materials (4.3%). Notably, the ten most frequent space groups account for 11,796 materials, which is nearly half of the entire dataset (49.1%), as illustrated in Figure 3.1. This distribution aligns with the statistics reported by Baur and Kassner, confirming that the filtered dataset serves as a representative subset of the known inorganic chemical space.

Not all crystallographic space groups are represented in our dataset of prototypes. Of the 230 possible space groups, only 199 occur among the ground-state structures from the Materials Project, meaning that 31 space groups are absent. This observation is significant as it quantifies the reduction in structural diversity imposed by the ground-state stability criterion. The missing groups are: 16, 17, 22, 25, 48, 77, 89, 93, 94, 95, 101, 104, 106, 111, 153, 168, 169, 170, 171, 172, 177, 178, 179, 195, 196, 207, 208, 209, 210, 211, and 222. The absence of these groups can be explained by several factors. First, not all space groups are equally common in inorganic crystal chemistry; in fact, the ICSD has no entry for a material that belongs to space group $P4_222$ (No. 93). Second, our dataset is restricted to the ground-state phases of materials, which excludes many polymorphs stabilized at elevated temperature or pressure. As a result, space groups that are more typical of metastable or high-temperature polymorphs do not appear here.

This exclusion of high-temperature phases also explains the divergence in the most frequently occurring space groups when comparing the ten most prevalent groups in the present dataset (Table 3.1) to the historical ICSD data. While both analyses identify the exact same two most common space groups with similar relative percentages, their subsequent rankings differ significantly. Baur and Kassner reported high-symmetry cubic groups such as $Fm\bar{3}m$ (No. 225, 4.4%) and $Fd\bar{3}m$

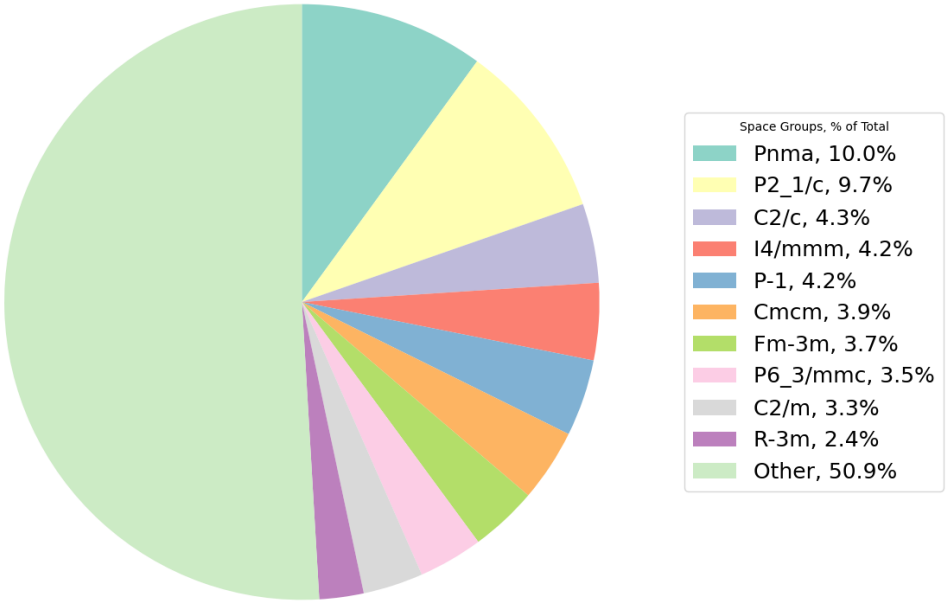


Figure 3.1: Most frequent space groups of all ground state materials.

(No. 227, 3.0%) among the most frequent. In contrast, these cubic groups are markedly less prominent in the present ground-state dataset, which instead exhibits an enhanced representation of lower-symmetry groups such as *C*2/*c* and *Cmcm*. A clear example of this phenomenon is the perovskite structure type. The "classic" perovskite structure is the cubic phase that exists at high temperatures for many ABX_3 materials, whereas their thermodynamic ground state typically consists of a lower-symmetry structure [76, 77]. Consequently, while these high-symmetry structures are prevalent in the full ICSD, they are systematically down-weighted in the results of this study.

The structural complexities of this perovskite family provide an excellent demonstration of why a geometry-driven, species-agnostic framework comparison is necessary for identifying prototypes. As first classified by Glazer [78, 79], the ideal cubic perovskite framework can accommodate these ground-state structural distortions through the cooperative tilting of its corner-sharing

octahedra. These geometric rotations give rise to 23 distinct tilt systems, each manifesting in a different, lower-symmetry crystallographic space group. Algorithms that rigidly enforce symmetry constraints or exact Wyckoff site mappings often fail to recognize the fundamental relationship between these variants, classifying them as entirely separate prototypes. By utilizing a framework comparator with dynamically tuned geometric tolerances, the algorithm can overlook these symmetric reductions and correctly identify the shared topological backbone connecting the entire family of distorted perovskite ground states, as will be shown later.

Table 3.1: Top ten space groups of ground state materials compared with those reported by Baur and Kassner.

This Work (Materials Project)				**Baur & Kassner (1992, ICSD)**			
Space Group	SG Number	Count	%	Space Group	SG Number	Count	%
Pnma	62	2,321	10.0	*Pnma*	62	2,863	8.3
$P2_1/c$	14	2,235	9.7	$P2_1/c$	14	2,827	8.2
C2/c	15	995	4.3	$Fm\bar{3}m$	225	1,532	4.4
I4/mmm	139	968	4.2	$P\bar{1}$	2	1,508	4.4
$P\bar{1}$	2	964	4.2	*C2/c*	15	1,326	3.8
Cmcm	63	911	3.9	$P6_3/mmc$	194	1,254	3.6
$Fm\bar{3}m$	225	847	3.7	*C2/m*	12	1,180	3.4
$P6_3/mmc$	194	810	3.5	*I4/mmm*	139	1,176	3.4
C2/m	12	756	3.3	$Fd\bar{3}m$	227	1,050	3.0
$R\bar{3}m$	166	557	2.4	$R\bar{3}m$	166	858	2.5
Top 10 total	-	**11,796**	**49.1**	**Top 10 total**	-	**15,574**	**44.9**

Together, these considerations show that while our dataset provides a broad and representative view of crystallographic symmetry in stable inorganic compounds, it does not exhaustively cover the entire space-group spectrum. This highlights the complementarity between ground-state computational databases and experimental compilations such as the ICSD: the former capture thermodynamic stability trends, while the latter provide a more complete inventory of reported structures, including metastable and exotic cases.

3.1 The Effect of Tolerances on the Number of Structure Prototypes

Deciding the tolerance values for this project was a crucial step before going ahead and identifying structure prototypes. A loose tolerance means that the matching criterion would be more lenient, and a strict tolerance means a harsher matching criterion. Therefore, the looser the tolerance (the higher its value), the more materials would fit each candidate framework, and *vice versa*. In this section results and affects of varying each of the tolerances will be discussed, along with the conclusion for the values chosen for the continuation of this research.

The results shown in Figure 3.2 show the number of frameworks as a function of length tolerance conform to the expectation of a large number of prototypes for strict tolerance, which

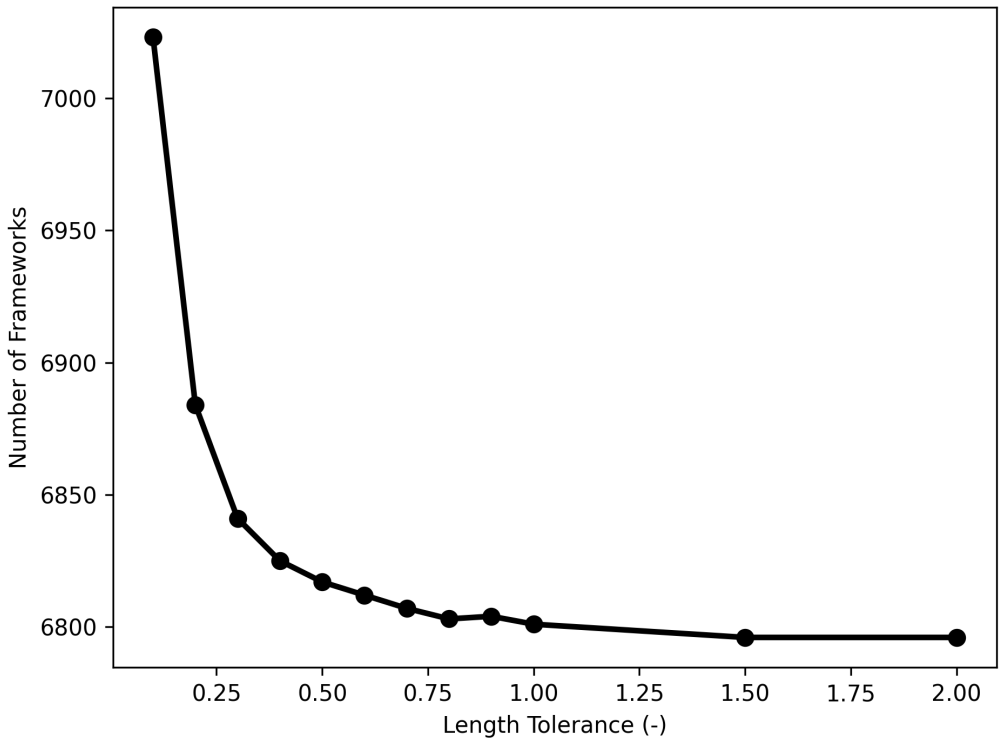


Figure 3.2: Number of distinct prototypes identified as a function of length tolerance.

decreases as the length tolerance becomes looser. It could also be observed that for values higher than one, there is no significant change in number of prototypes.

The results shown in Figure 3.2 illustrate the number of identified frameworks as a function of length tolerance. As expected, a strict tolerance yields a superficially large number of prototypes, because minor geometric variations (such as computational relaxation differences or physical thermal expansion) cause identical topological frameworks to be classified as distinct. As the tolerance becomes looser, the algorithm successfully groups these variations, leading to a sharp decrease in the total number of prototypes. However, for $ltol$ values greater than one, the curve reaches a plateau with no significant change in the number of prototypes. Physically, an $ltol$ of 1.0 implies that a 100% relative difference in lattice vector lengths is permitted, effectively removing the length constraint entirely. In this regime, the number of prototypes is governed solely by the remaining angle and site constraints, demonstrating that further relaxing the length tolerance provides no additional topological grouping.

The results shown in Figure 3.3 of number of frameworks as a function of angle tolerance show similar trend to the one in Figure 3.2 with the biggest difference being lack of a clear plateau. The reason for this is that for high values of angle tolerance, a high number of possible mappings are checked. It leads to a long runtime, which caused some calculations to be timed out. Ultimately,

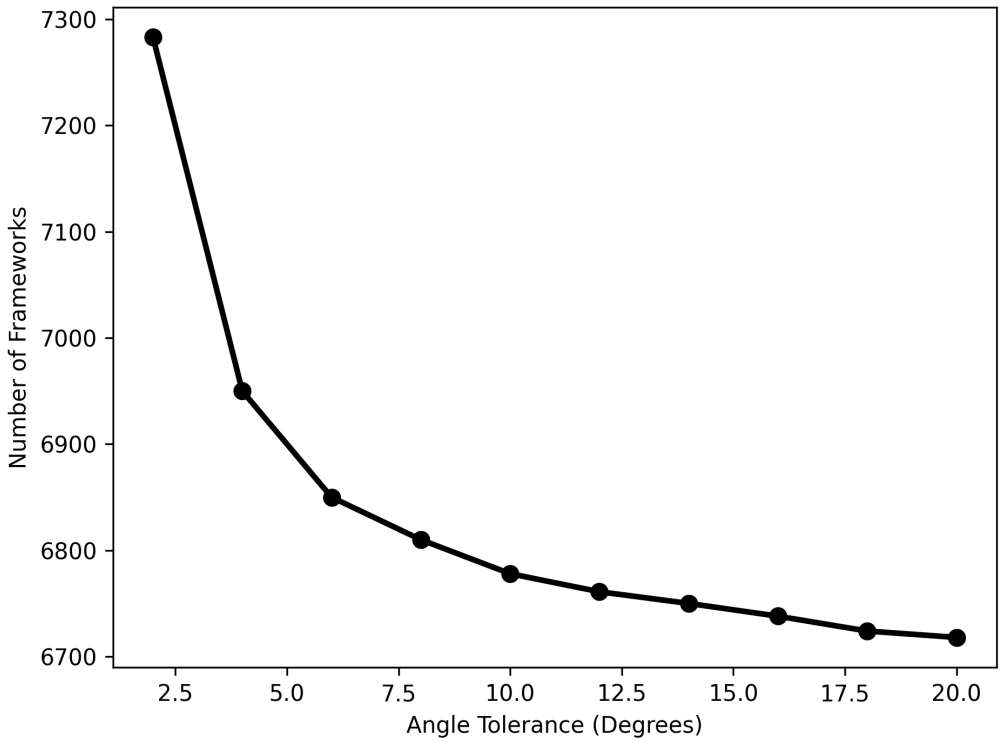


Figure 3.3: Number of distinct prototypes identified as a function of angle tolerance.

it was decided that the set of values shown here $(2-20^{\circ})$ is sufficient to observe the influence of varying values for angle tolerance and decide on a reasonably suitable value of angle tolerance.

The results for the number of frameworks as a function of site tolerance show a slightly different shape compared to the length and angle tolerances, though they follow the same general trend where the total number of identified prototypes decreases as the tolerance becomes looser. Here, a clear intermediate plateau can be seen bounded by steep drops. This plateau represents the optimal matching region. At low tolerance values before the plateau, the algorithm under-fits the data. In this regime, the criteria are overly strict, causing physically identical frameworks with minor atomic displacements (such as those arising from chemical substitution or slight structural relaxations) to be incorrectly classified as separate prototypes. Conversely, the steep drop following the plateau indicates severe over-fitting. As the site tolerance becomes too loose, the algorithm begins merging fundamentally distinct structures into single prototypes. Ultimately, as the value approaches $stol = 1$, the physical location of atoms inside the unit cell becomes mathematically irrelevant. At this extreme, structures are matched based almost entirely on their macroscopic symmetry and the number of atomic sites, leading to an artificially collapsed number of frameworks. Therefore, unlike the length and angle tolerances, the site tolerance requires careful calibration to sit squarely within this intermediate plateau.

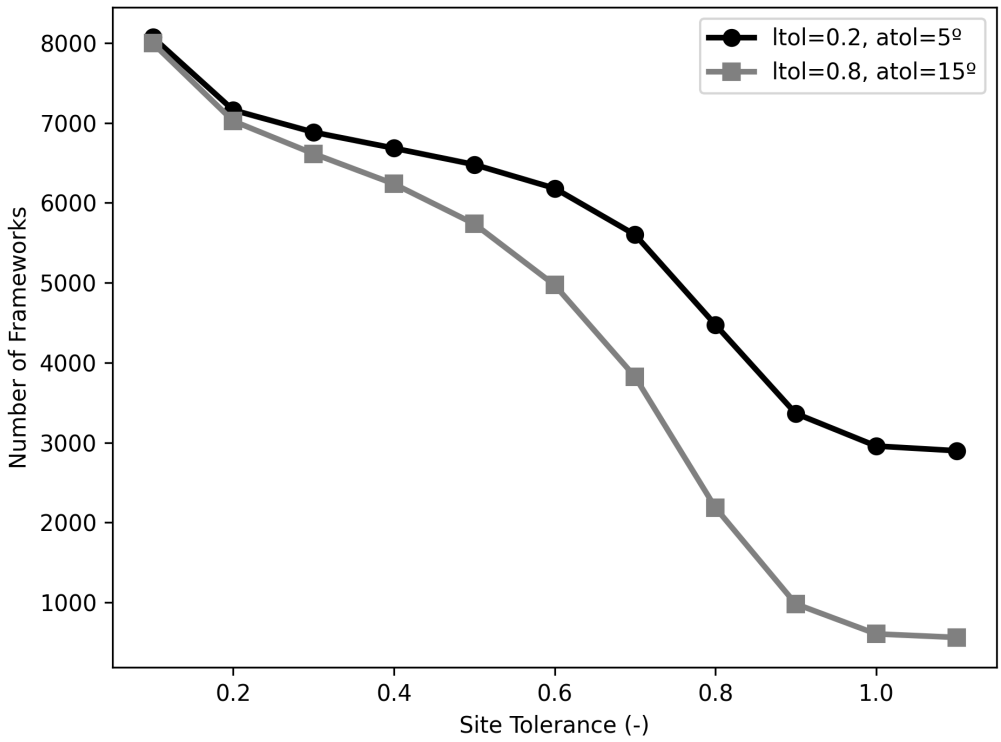


Figure 3.4: Number of distinct prototypes identified as a function of site tolerance with two sets of different values for length and angle tolerances - a strict set and loose one.

Since tolerances criteria impact the number of resultant prototypes, they also affect the number of matching operations that must be made. Since every material is matched with a large list of frameworks, the size of this list is directly related to the number of interactions.

After careful considerations, the tolerances that were chosen are: a length tolerance of 0.2, an angle tolerance of 5 degrees, and a site tolerance of 0.3. This choice serves as a compromise between over-fitting, thus giving too many different frameworks even though they appear the same, and under-fitting, thus matching every framework even when they should be different.

3.2 Structure Prototype Identification

Applying the matching methodology described in section 2.2 with the tolerances just mentioned yielded 6,898 unique frameworks. Of these, 2,410 matched two or more materials and were consequently labeled as prototypes. The distribution of these identified prototypes shows that a select few structural forms represent a significant proportion of crystalline materials. Notably, the five most frequent prototypes account for 7.9% of the database structures. Leading the list is the intermetallic $CaAl_4$ structure with 565 matching entries, crystallizing in the tetragonal space group

$I4/mmm$ (No. 139). It is followed by the CaMgSi prototype with 521 matches, an intermetallic structure in space group *Pnma* (No. 62) that accommodates both ABC and AB_2 stoichiometries. The third most common framework is Ti_6Si_2B with 294 matches, which adopts the hexagonal $P\bar{6}2m$ (No. 189) symmetry and describes AB_2C_6, AB_4C_4, and AB_8 stoichiometries. Finally, the fourth and fifth positions are occupied by Ca_3SiO and $CaFeO_3$, each with 223 matches. These correspond to the anti-perovskite (AB_3X) and distorted perovskite (ABX_3) frameworks, with space groups $I4/mmm$ (No. 139) and *Pnma* (No. 62), respectively. The high frequency of the $CaFeO_3$ prototype directly reflects the algorithm's ability to successfully group the diverse tilted octahedral systems discussed previously.

To contextualize the distribution of structure prototypes, it is instructive to examine the statistical analyses of intermetallic compounds conducted by Steurer and Dshemuchadse [80, 81]. Their extensive review mapped the frequency of various structure types to understand the overarching crystallographic landscape, revealing that a small fraction of prototypes accounts for a disproportionately large share of known materials. However, their methodological approach differs fundamentally from the one employed in this research. Their analysis relies heavily on historical structure type assignments curated within Pearson's Crystal Data database [82], many of which were originally identified by analogy using powder X-ray diffraction patterns rather than direct geometric comparison. In contrast, this thesis utilizes a systematic computational workflow using the FrameworkComparator to group structures based strictly on atomic topology, completely independent of predefined database metadata or chemical species.

Despite these differing methodologies, both studies converge on remarkably similar physical conclusions regarding structural distribution and symmetry. Beyond the shared observation of a highly concentrated prototype landscape, both works identify an inherent statistical bias regarding low-symmetry structures. Just as Steurer and Dshemuchadse noted that lower-symmetry structure types are inherently less flexible to atomic substitution, the present work demonstrates that the greater degrees of freedom in triclinic and monoclinic systems make them highly sensitive to minor geometric variations. This sensitivity prevents low-symmetry frameworks from being matched as effectively as their high-symmetry counterparts under uniform geometric tolerances.

Based on the initial distribution results, the dataset was further refined by applying the additional filters and stoichiometric splitting detailed in the Methods chapter. In this phase, the matching process was restricted so that materials were compared only to those sharing the same stoichiometry; for instance, Al_2O_3 was matched exclusively with other materials belonging to the A_2X_3 class. This approach yields a distinct list of prototypes for every stoichiometry group, wherein each prototype retains explicit information regarding the atomic positions of specific ion species.

In total, the materials were partitioned into 174 distinct stoichiometry groups. The eight most frequent groups are shown in Figure 3.6. Within each of these subsets, the matching process followed the same procedure as illustrated in Figure 2.3.

Observing the results of the most frequent groups (Figure 3.6), it can be observed that they are

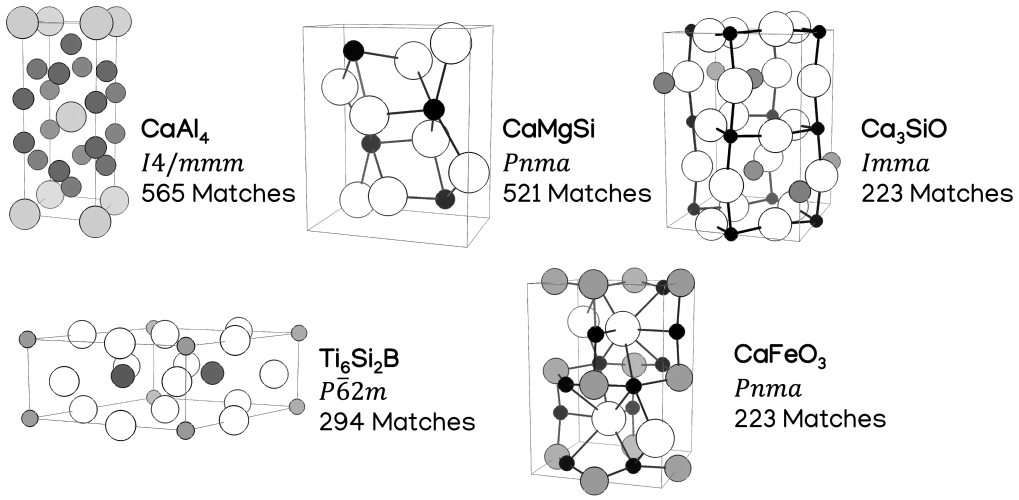


Figure 3.5: Most popular structure prototypes of ground state materials.

relatively simple in their stoichiometry. All of them contain only one species of anion, and most contain only one or two kinds of cations. In addition, the number of cations and anions is relatively small. In contrast, from a combinatorial point of view, the number of possible combinations grows with the increase in stoichiometric complexity. The reason for the prevalence of simple stoichiometries is probably rooted in thermodynamic causes. For example, a compound with a stoichiometry of ABC_2XY_4 could decompose into two distinct compounds with stoichiometries AB_2X_4 and AX. On the other hand, stoichiometries that have a moderate degree of complexity contain significantly more materials than the simplest groups. For example, AB_2X_4 contains more than twice the number of materials as AX.

In total, the analysis yielded 2,073 unique frameworks. Of these, 697 matched two or more materials within their stoichiometry group and were labeled as prototypes, with 238 matching five or more. Notably, the five most frequent prototypes account for 10.3% of the structures in the filtered database. The most popular is the $NaFeO_2$ prototype with 165 matching structures, crystallizing in the space group $R\bar{3}m$ (No. 166). This is followed by the CaO prototype with 122 matches, which corresponds to the rock-salt structure in space group $Fm\bar{3}m$ (No. 225). The third most common is $CaFeO_3$ with 100 matches, which represents the *Pnma* (No. 62) distorted perovskite framework resulting from the octahedral tilting mechanisms discussed earlier in this chapter. The fourth is $NaZrCuS_3$ with 95 matches in space group *Cmcm* (No. 63), and the fifth is the inverse perovskite $AlFe_3C$ with 92 matches in space group $Pm\bar{3}m$ (No. 221). Unit cells of these prototypes are visualized in Figure 3.7, and the full list is provided in the supplementary information.

The pattern of frequent prototypes accounting for a high share of materials can be observed within stoichiometry groups. For example, the three most frequent prototypes in stoichiometry groups AB_2X_4, ABX_2, and ABX_3 account for 41%, 46%, and 42% of the total materials, respec-

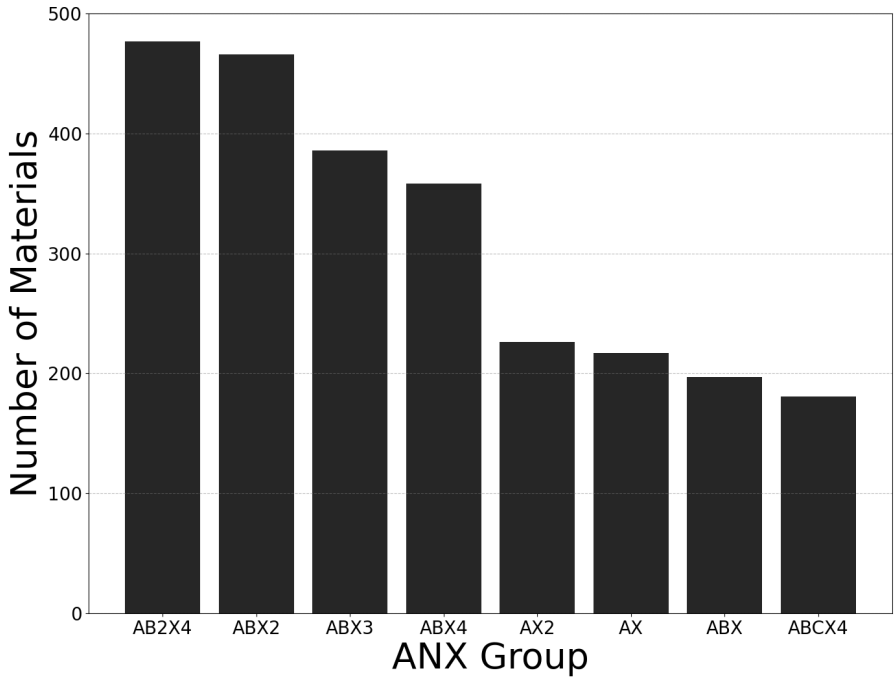


Figure 3.6: Top 8 stoichiometry groups ranked by number of constituent materials.

tively. This recurrence confirms a general principle: the vast majority of known structures are represented by a relatively small number of prototypes. This finding is particularly relevant for structure prediction, as it suggests that a compact set of 'seed' structures can provide high coverage of the plausible configuration space for undiscovered materials.

An additional observation is that the most frequent structure prototypes largely coincide with the most frequent space groups. However, there is a notable absence of triclinic and monoclinic systems among the most common prototypes. This discrepancy can be attributed to symmetry constraints: low-symmetry systems possess a large number of degrees of freedom regarding lattice and atomic positional parameters. Consequently, small structural variations often lead to mismatches, preventing these materials from clustering as consistently as those in higher-symmetry groups. Conversely, high-symmetry space groups impose strict geometric restrictions, increasing the likelihood that different compounds map onto a single prototype. This trend is reinforced by the `StructureMatcher` algorithm, which distinguishes structures based on precise lattice metrics. As a result, prototype statistics favor higher-symmetry systems, while triclinic and monoclinic frameworks are fragmented into many small clusters despite their prevalence in the raw space-group distribution. Although this fragmentation could be mitigated by relaxing the matching tolerances, strict parameters were maintained to ensure consistency with ICSD classification

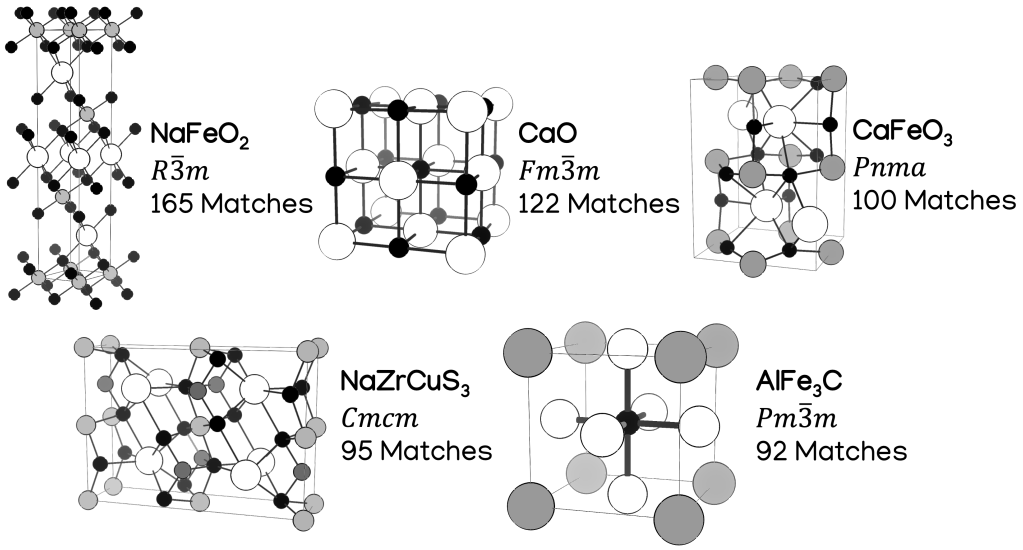


Figure 3.7: Most popular prototypes among the filtered ground state database.

criteria, particularly regarding c/a ratios and β angle ranges.

Collectively, these analyses demonstrate that the vast chemical space of inorganic materials can be effectively mapped onto a finite and manageable set of structural prototypes. Despite the inherent challenges in clustering low-symmetry systems, the identification of 697 robust prototypes, covering the majority of stoichiometric and symmetry variations, provides a high-fidelity discrete basis for structural classification. By reducing the infinite possibilities of atomic arrangements into a curated library of "seed" structures, this work establishes a foundational framework for data-driven structure prediction. The high coverage of these prototypes suggests that they can serve as reliable starting points for high-throughput computational screening, substantially narrowing the search space for discovering new stable materials.

4 Conclusions

In this work, a systematic process for characterizing unique structural prototypes from a large database of experimental materials was presented. The primary goal of this research was to create a reliable, experimentally grounded resource of structure prototypes based on thermodynamic ground-state materials, which could be utilized to enhance modern materials discovery efforts. The research utilized a dataset of 23,160 experimental ground-state materials from the Materials Project database. Structural comparison was performed using the `pymatgen` library and its `StructureMatcher` module, treating atoms as topologically equivalent sites to identify shared geometric frameworks regardless of chemical composition. Following a systematic investigation of comparison thresholds, the specific tolerances defining a structural match were established as $ltol = 0.2$, $atol = 5^\circ$, and $stol = 0.3$. The dataset was subsequently filtered and sorted into 174 stoichiometric (ANX) groups, leading to the identification of 2,073 unique structural *frameworks*, of which 697 were defined as *prototypes* matching two or more materials.

The most significant finding emerging from this classification is the observation that the inorganic structural landscape is dominated by a relatively small number of unique prototypes. A small number of unique prototypes accounts for a disproportionately large share of known materials. In the filtered dataset, the top five prototypes alone represent over 10% of all structures. This implies that materials discovery and structure prediction algorithms can be made significantly more efficient by prioritizing this limited set of "seed" structures, as they cover a vast region of the thermodynamically stable structural landscape. Furthermore, the results demonstrate a clear inverse relationship between stoichiometric complexity and material abundance. The vast majority of ground-state materials crystallize in simple stoichiometries (e.g., ABX_3, AB_2X_4), while complex compositions are rare. This indicates that despite the increasing combinatorial possibilities associated with complex chemical formulas, the number of realized stable prototypes in nature remains concentrated within these simpler stoichiometric families.

A comparison between the raw dataset and the identified prototypes reveals a complex relationship regarding crystal symmetry. In some cases, the trends align; for instance, the orthorhombic space group *Pnma*, which is the most frequent symmetry in the raw dataset, remains a dominant symmetry among the identified prototypes. However, a significant divergence appears regarding lower-symmetry systems. While triclinic and monoclinic space groups (such as $P2_1/c$ and $P\bar{1}$) represent a large portion of the experimental data, they are notably absent from the most com-

mon structure prototypes. This discrepancy indicates that the matching algorithm exhibits a bias against low-symmetry structures. Because such lattices possess greater degrees of freedom, having variable lattice parameters and angles, compared to cubic or tetragonal systems, they are more sensitive to minor geometric variations. Consequently, geometrically similar low-symmetry structures are less likely to satisfy the strict matching tolerances, preventing them from clustering as effectively as their high-symmetry counterparts.

However, this work has several limitations. First, the proposed comparison method is computationally expensive because it matches structures directly through their atomic configurations. In practice, the same crystal structure can be represented by many equivalent choices of unit cells, so the matching procedure must search over cell transformations (and, in some cases, reduced-cell representations), which increases the computational cost. Second, the prototype-assignment procedure is order-dependent because each prototype is defined relative to a representative structure that is updated during sequential matching. The tolerances allow two slightly different structures to be considered a match, the rule used to choose or update the representative can introduce a cumulative bias. In particular, because the representative is selected as the matched structure with the larger unit cell volume, the representative can "grow" over time as additional matches are incorporated. As a result, a structure that would match the initial representative may fail to match the final one, even though the two are connected by a sequence of tolerance-accepted intermediate matches. For example, for a tetragonal prototype, repeatedly selecting the larger-volume match can gradually change the representative lattice parameters and shift the axial ratio c/a, ultimately making the final representative incompatible with early members of the group under the same matching tolerances. In practice, this effect primarily causes small fluctuations in the number of materials assigned to each prototype, while the overall set of identified prototypes remains unchanged.

This work could be further expanded by developing a dedicated prototype-assignment and analysis framework for metallic and intermetallic compounds. Although the comparison in this study was performed across the full set of ground state crystalline materials, the focused analysis and prototype matching strategy were tailored to ionic solids, which contain a clear cation-anion arrangement. For metallic systems, where a strict cation–anion arrangement does not fit, a separate methodology to define and assign prototypes is required.

The full lists of prototypes is freely available on GitHub[1]. Each prototype is shown with the following information: filename (as obtained from the Materials Project), chemical formula, how many materials matched this prototype, space group (both in Hermann-Mauguin and IUCr notation), orientation matrix (in Ångström), and fractional coordinates for each atom.

In summary, this work provides an automated, reproducible process for generating a well-defined library of structural prototypes. By curating this collection from thermodynamically stable ground states, this thesis offers a critical, data-driven foundation for next-generation computational tools in materials discovery.

[1] link: https://github.com/bmd-lab/structure-prototypes

References

[1] Van de Walle, A. A complete representation of structure–property relationships in crystals. *Nature materials* **2008**, *7*, 455–458.

[2] Cheng, Y. Q.; Ma, E. Atomic-level structure and structure–property relationship in metallic glasses. *Progress in Materials Science* **2011**, *56*, 379–473.

[3] Wang, X.; Faizan, M.; Fu, Y.; Zhou, K.; Zhang, Y.; He, X.; Singh, D. J.; Zhang, L. Influence of Local Cation Order on Electronic Structure and Optical Properties of Cation-Disordered Semiconductor AgBiS2. *Chinese Physics Letters* **2024**, *41*, 106101.

[4] Slack, G. A. Thermal Conductivity of Pure and Impure Silicon, Silicon Carbide, and Diamond. *Journal of Applied Physics* **1964**, *35*, 3460–3466.

[5] Saslow, W.; Bergstresser, T.; Cohen, M. L. Band structure and optical properties of diamond. *Physical Review Letters* **1966**, *16*, 354.

[6] Sánchez Egea, A. J.; Martynenko, V.; Abate, G.; Deferrari, N.; Martinez Krahmer, D.; López de Lacalle, L. N. Friction capabilities of graphite-based lubricants at room and over 1400 K temperatures. *The International Journal of Advanced Manufacturing Technology* **2019**, *102*, 1623–1633.

[7] Blakslee, O. L.; Proctor, D. G.; Seldin, E. J.; Spence, G. B.; Weng, T. Elastic Constants of Compression-Annealed Pyrolytic Graphite. *Journal of Applied Physics* **1970**, *41*, 3373–3382.

[8] Calzaferri, G.; Rytz, R. The band structure of diamond. *The Journal of Physical Chemistry* **1996**, *100*, 11122–11124.

[9] Slonczewski, J.; Weiss, P. Band structure of graphite. *Physical review* **1958**, *109*, 272.

[10] Painter, G.; Ellis, D. Electronic band structure and optical properties of graphite from a variational approach. *Physical Review B* **1970**, *1*, 4747.

[11] Yankowitz, M.; Chen, S.; Polshyn, H.; Zhang, Y.; Watanabe, K.; Taniguchi, T.; Graf, D.; Young, A. F.; Dean, C. R. Tuning superconductivity in twisted bilayer graphene. *Science* **2019**, *363*, 1059–1064.

[12] Cao, Y.; Fatemi, V.; Fang, S.; Watanabe, K.; Taniguchi, T.; Kaxiras, E.; Jarillo-Herrero, P. Unconventional superconductivity in magic-angle graphene superlattices. *Nature* **2018**, *556*, 43–50.

[13] Bao, C.; Yao, W.; Wang, E.; Chen, C.; Avila, J.; Asensio, M. C.; Zhou, S. Stacking-dependent electronic structure of trilayer graphene resolved by nanospot angle-resolved photoemission spectroscopy. *Nano letters* **2017**, *17*, 1564–1568.

[14] Zhang, J.; Xiao, J.; Meng, X.; Monroe, C.; Huang, Y.; Zuo, J.-M. Free folding of suspended graphene sheets by random mechanical stimulation. *Physical review letters* **2010**, *104*, 166805.

[15] Cui, T.; Mukherjee, S.; Cao, C.; Sudeep, P. M.; Tam, J.; Ajayan, P. M.; Singh, C. V.; Sun, Y.; Filleter, T. Effect of lattice stacking orientation and local thickness variation on the mechanical behavior of few layer graphene oxide. *Carbon* **2018**, *136*, 168–175.

[16] Kittel, C. *Introduction to Solid State Physics*, 8th ed.; Wiley: New York, 2005.

[17] Fan, Q.; Chai, C.; Wei, Q.; Yan, H.; Zhao, Y.; Yang, Y.; Yu, X.; Liu, Y.; Xing, M.; Zhang, J.; others Novel silicon allotropes: Stability, mechanical, and electronic properties. *Journal of Applied Physics* **2015**, *118*.

[18] Giesecke, G.; Pfister, H. Präzisionsbestimmung der Gitterkonstanten von AIIIBv-Verbindungen. *Acta Crystallographica* **1958**, *11*, 369–371.

[19] Sze, S. M.; Ng, K. K. *Physics of Semiconductor Devices*, 3rd ed.; John Wiley & Sons: Hoboken, NJ, 2007.

[20] Putnis, A. *An Introduction to Mineral Sciences*; Cambridge University Press, 1992.

[21] Mehl, M.; Hicks, D.; Toher, C.; Levy, O.; Hanson, R.; Hart, G.; Curtarolo, S. The AFLOW Library of Crystallographic Prototypes: Part 1. *United States Navy: Publications* **2017**,

[22] Ewald, P. P., Hermann, C., Eds. *Strukturbericht 1913–1928*; Zeitschrift für Kristallographie - Ergänzungsband; Akademische Verlagsgesellschaft: Leipzig, 1931; Vol. 1.

[23] Mehl, M. J. A brief history of strukturbericht symbols and other crystallographic classification schemes. Journal of Physics: Conference Series. 2019; p 012016.

[24] Pearson, W. B. *A Handbook of Lattice Spacings and Structures of Metals and Alloys*; International Series of Monographs on Metal Physics and Physical Metallurgy 4; Pergamon Press: Oxford, 1958; Vol. 1; Reprinted by Elsevier (2013).

[25] Villars, P.; Calvert, L. D. *Pearson's Handbook of Crystallographic Data for Intermetallic Phases*, 2nd ed.; ASM International: Materials Park, OH, 1991.

[26] Curtarolo, S.; Setyawan, W.; Hart, G. L.; Jahnatek, M.; Chepulskii, R. V.; Taylor, R. H.; Wang, S.; Xue, J.; Yang, K.; Levy, O.; others AFLOW: An automatic framework for high-throughput materials discovery. *Computational Materials Science* **2012**, *58*, 218–226.

[27] Materials Data on CaO by Materials Project.

[28] Magee, C. L. The role of materials innovation in overall technological development. *JOM* **2012**, *64*, 536–543.

[29] Moran, M. J.; Shapiro, H. N.; Boettner, D. D.; Bailey, M. B. *Fundamentals of Engineering Thermodynamics*, 8th ed.; John Wiley & Sons, 2014.

[30] Reed, R. C. *The Superalloys: Fundamentals and Applications*; Cambridge University Press, 2008.

[31] Pollock, T. M.; Tin, S. Nickel-based superalloys for advanced turbine engines: chemistry, microstructure and properties. *Journal of Propulsion and Power* **2006**, *22*, 361–374.

[32] Ceder, G.; Morgan, D.; Fischer, C.; Tibbetts, K.; Curtarolo, S. Opportunities and challenges for a first-principles materials database. *MRS Bulletin* **2011**, *36*, 984–991.

[33] Sun, W.; Dacek, S. T.; Ong, S. P.; Hautier, G.; Jain, A.; Richards, W. D.; Gamst, A. C.; Persson, K. A.; Ceder, G. The Thermodynamic Scale of Inorganic Crystalline Metastability. *Science Advances* **2016**, *2*, e1600225.

[34] Curtarolo, S.; Hart, G. L. W.; Nardelli, M. B.; Mingo, N.; Sanvito, S.; Levy, O. The high-throughput highway to computational materials design. *Nature Materials* **2013**, *12*, 191–201.

[35] Jain, A.; Ong, S. P.; Hautier, G.; Chen, W.; Richards, W. D.; Dacek, S.; Cholia, S.; Gunter, D.; Skinner, D.; Ceder, G.; Persson, K. A. Commentary: The Materials Project: A materials genome approach to accelerating materials innovation. *APL Materials* **2013**, *1*, 011002.

[36] Alberi, K.; Nardelli, M. B.; Zakutayev, A.; Mitas, L.; Curtarolo, S.; Jain, A.; Fornari, M.; Marzari, N.; Takeuchi, I.; Green, M. L.; others The 2019 materials by design roadmap. *Journal of Physics D: Applied Physics* **2018**, *52*, 013001.

[37] Pickard, C. J.; Needs, R. J. Ab initio random structure searching. *Journal of Physics: Condensed Matter* **2011**, *23*, 053201.

[38] Oganov, A. R.; Glass, C. W. Crystal structure prediction using ab initio evolutionary techniques: Principles and applications. *The Journal of Chemical Physics* **2006**, *124*, 244704.

[39] Merchant, A.; Batzner, S.; Schoenholz, S. S.; Aykol, M.; Cheon, G.; Cubuk, E. D. Scaling deep learning for materials discovery. *Nature* **2023**, *624*, 80–85.

[40] Cheetham, A. K.; Seshadri, R. Artificial intelligence driving materials discovery? perspective on the article: Scaling deep learning for materials discovery. *Chemistry of Materials* **2024**, *36*, 3490–3495.

[41] Leeman, J.; Liu, Y.; Stiles, J.; Lee, S. B.; Bhatt, P.; Schoop, L. M.; Palgrave, R. G. Challenges in high-throughput inorganic materials prediction and autonomous synthesis. *PRX Energy* **2024**, *3*, 011002.

[42] 'crystallinity' in IUPAC Compendium of Chemical Terminology, 5th ed. International Union of Pure and Applied Chemistry; 2025. Online version 5.0.0, 2025. https://doi.org/10.1351/goldbook.C01433. **2025**,

[43] De Graef, M.; McHenry, M. E. *Structure of materials : an introduction to crystallography, diffraction, and symmetry*, second edition, fully revised and updated ed.; Cambridge University Press: New York, 2012.

[44] Aroyo, M. I., Ed. *International Tables for Crystallography, Volume A: Space-group symmetry*, 6th ed.; International Union of Crystallography: Chester, UK, 2016.

[45] Bravais, A. Mémoire sur les systèmes formés par les points distribués régulièrement sur un plan ou dans l'espace. *Journal de l'École Polytechnique* **1850**, *19*, 1–128.

[46] Wyckoff, R. W. G. *The Analytical Expression of the Results of the Theory of Space Groups*; Carnegie Institution of Washington, 1922.

[47] Wyckoff, R. W. G. *Crystal Structures, Vol. 1*, 2nd ed.; Interscience: New York, 1963.

[48] IUCr Wyckoff position — Online Dictionary of Crystallography. `https://dictionary.iucr.org/Wyckoff_position`, 2017.

[49] Zagorac, D.; Müller, H.; Ruehl, S.; Zagorac, J.; Rehme, S. Recent developments in the Inorganic Crystal Structure Database: theoretical crystal structure data and related features. *Applied Crystallography* **2019**, *52*, 918–925.

[50] Allmann, R.; Hinek, R. The introduction of structure types into the Inorganic Crystal Structure Database ICSD. *Foundations of Crystallography* **2007**, *63*, 412–417.

[51] Steudel, D. A.; Rühl, D. S.; Hinek, D. R.; Rehme, S. Scientific Manual ICSD Database.

[52] Ong, S. P.; Richards, W. D.; Jain, A.; Hautier, G.; Kocher, M.; Cholia, S.; Gunter, D.; Chevrier, V. L.; Persson, K. A.; Ceder, G. Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis. *Computational Materials Science* **2013**, *68*, 314–319.

[53] Hicks, D.; Toher, C.; Ford, D. C.; Rose, F.; Santo, C. D.; Levy, O.; Mehl, M. J.; Curtarolo, S. AFLOW-XtalFinder: a reliable choice to identify crystalline prototypes. *npj Computational Materials* **2021**, *7*, 30.

[54] Lonie, D. C.; Zurek, E. Identifying duplicate crystal structures: XTALCOMP, an open-source solution. *Computer Physics Communications* **2012**, *183*, 690–697.

[55] Gelato, L. M.; Parthé, E. STRUCTURE TIDY a computer program to standardize crystal structure data. *Journal of Applied Crystallography* **1987**, *20*, 139–143.

[56] Dzyabchenko, A. V. Method of crystal-structure similarity searching. *Acta Crystallographica Section B* **1994**, *50*, 414–425.

[57] Hundt, R.; Schön, J. C.; Jansen, M. CMPZ an algorithm for the efficient comparison of periodic structures. *Journal of Applied Crystallography* **2006**, *39*, 6–16.

[58] Su, C.; Lv, J.; Li, Q.; Wang, H.; Zhang, L.; Wang, Y.; Ma, Y. Construction of crystal structure prototype database: methods and applications. *Journal of Physics: Condensed Matter* **2017**, *29*, 165901.

[59] Flor, G.; Orobengoa, D.; Tasci, E.; Perez-Mato, J. M.; Aroyo, M. I. Comparison of structures applying the tools available at the Bilbao Crystallographic Server. *Applied Crystallography* **2016**, *49*, 653–664.

[60] Aroyo, M. I.; Perez-Mato, J. M.; Orobengoa, D.; Tasci, E.; de la Flor, G.; Kirov, A. Crystallography online: Bilbao crystallographic server. *Bulg. Chem. Commun* **2011**, *43*, 183–197.

[61] Aroyo, M. I.; Perez-Mato, J. M.; Capillas, C.; Kroumova, E.; Ivantchev, S.; Madariaga, G.; Kirov, A.; Wondratschek, H. Bilbao Crystallographic Server: I. Databases and crystallographic computing programs. *Zeitschrift für Kristallographie-Crystalline Materials* **2006**, *221*, 15–27.

[62] Aroyo, M. I.; Kirov, A.; Capillas, C.; Perez-Mato, J.; Wondratschek, H. Bilbao Crystallographic Server. II. Representations of crystallographic point groups and space groups. *Foundations of Crystallography* **2006**, *62*, 115–128.

[63] Tasci, E.; de La Flor, G.; Orobengoa, D.; Capillas, C.; Perez-Mato, J.; Aroyo, M. An introduction to the tools hosted in the Bilbao Crystallographic Server. EPJ Web of Conferences. 2012; p 00009.

[64] Togo, A.; Shinohara, K.; Tanaka, I. Spglib: a software library for crystal symmetry search. *Science and Technology of Advanced Materials: Methods* **2024**, *4*, 2384822.

[65] Niggli, P. *Krystallographische und strukturtheoretische Grundbegriffe*; Akademische verlagsgesellschaft mbh, 1928; Vol. 1.

[66] Křivỳ, I.; Gruber, B. A unified algorithm for determining the reduced (Niggli) cell. *Foundations of Crystallography* **1976**, *32*, 297–298.

[67] Santoro, A.; Mighell, A. D. Determination of reduced cells. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography* **1970**, *26*, 124–127.

[68] Kuhn, H. W. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* **1955**, *2*, 83–97.

[69] Bartel, C. J.; Millican, S. L.; Deml, A. M.; Rumptz, J. R.; Tumas, W.; Weimer, A. W.; Lany, S.; Stevanović, V.; Musgrave, C. B.; Holder, A. M. Computational Materials Design: A Review on the Convex Hull of Stability. *Journal of Materials Science* **2022**, *57*, 10941–10968.

[70] Popov, I. V.; Görne, A. L.; Tchougréeff, A. L.; Dronskowski, R. Relative Stability of Diamond and Graphite as Seen Through Bonds and Hybridizations. *Physical Chemistry Chemical Physics* **2019**, *21*, 10961–10969.

[71] Zhao, W.; Pan, J.; Fang, Y.; Che, X.; Wang, D.; Bu, K.; Huang, F. Metastable MoS2: crystal structure, electronic band structure, synthetic approach and intriguing physical properties. *Chemistry–A European Journal* **2018**, *24*, 15942–15954.

[72] Cullity, B. D.; Stock, S. R. *Elements of X-ray Diffraction*, 3rd ed.; Prentice Hall, 2001.

[73] Ding, Y.; Kumagai, Y.; Oba, F.; Burton, L. A. Data-Mining Element Charges in Inorganic Materials. *The Journal of Physical Chemistry Letters* **2020**, *11*, 8264–8267.

[74] Haynes, W. M. *CRC handbook of chemistry and physics*; CRC press, 2016.

[75] Baur, W.; Kassner, D. The perils of Cc: comparing the frequencies of falsely assigned space groups with their general population. *Structural Science* **1992**, *48*, 356–369.

[76] Müller, K.; Berlinger, W.; Waldner, F. Characteristic structural phase transition in perovskite-type compounds. *Physical Review Letters* **1968**, *21*, 814.

[77] Kwei, G.; Lawson, A.; Billinge, S.; Cheong, S. Structures of the ferroelectric phases of barium titanate. *The Journal of Physical Chemistry* **1993**, *97*, 2368–2377.

[78] Glazer, A. M. The classification of tilted octahedra in perovskites. *Structural Science* **1972**, *28*, 3384–3392.

[79] Glazer, A. M. Simple ways of determining perovskite structures. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography* **1975**, *31*, 756–762.

[80] Dshemuchadse, J.; Steurer, W. Some statistics on intermetallic compounds. *Inorganic chemistry* **2015**, *54*, 1120–1128.

[81] Dshemuchadse, J.; Steurer, W. More statistics on intermetallic compounds–ternary phases. *Foundations of Crystallography* **2015**, *71*, 335–345.

[82] Villars, P.; Cenzual, K.; others Pearson's crystal data: crystal structure database for inorganic compounds. *(No Title)* **2007**,

A Appendix

A.1 Python Codes

Core python code used in this thesis.

A.1.1 compare-mat-framework.py

Listing 1

```python
import os
from pymatgen.core import Structure
from pymatgen.analysis.structure_matcher import StructureMatcher, FrameworkComparator
from abundance_rank import CompoundAbundanceScorer


def compare_frameworks(dir_path, abundance_excel_path, sheet_name=None):
    """
    Processes all .cif files in a single subdirectory, keeping the representative
    file with the largest volume for each unique framework and saving them in the same
    ↪ subdirectory.

    Parameters:
        dir_path (str): Path to the subdirectory to process.
    """
    unique_structures = []  # List to store tuples of (structure, file_path, cell_volume,
    ↪ count)

    # Initialize the structure matcher
    matcher = StructureMatcher(comparator=FrameworkComparator())

    scorer = CompoundAbundanceScorer(abundance_excel_path, sheet_name=sheet_name)

    # number of files processed
    count = 0

    # Loop over each .cif file in the subdirectory
    for file in sorted(os.listdir(dir_path)):
        if file.endswith(".cif"):
            count += 1
```

```
29
30              file_path = os.path.join(dir_path, file)
31
32              # Load the structure from the CIF file
33              structure = Structure.from_file(file_path)
34              primitive_structure = structure.get_primitive_structure()
35              reduced_formula = structure.composition.reduced_formula
36              abundance_score = float(scorer.calculate_compound_score(reduced_formula))
37              cell_volume = primitive_structure.volume
38
39
40              # Check if this structure matches any in the unique structures list
41              is_unique = True
42              for i, (rep_structure, rep_file, rep_abundance, abundant_structure, rep_volume,
                ↪  rep_count) in enumerate(unique_structures):
43                  if matcher.fit(primitive_structure, rep_structure):
44                      is_unique = False
45                      # Update the representative if the current structure has a larger volume
46                      if cell_volume > rep_volume:
47                          if abundance_score > rep_abundance:
48                              unique_structures[i] = (primitive_structure, file_path,
                                ↪  rep_abundance, primitive_structure, cell_volume,rep_count +
                                ↪  1)
49                          else:
50                              unique_structures[i] = (primitive_structure, file_path,
                                ↪  rep_abundance, abundant_structure, cell_volume, rep_count +
                                ↪  1)
51
52                      else:
53                          # Increment the count for this framework
54                          if abundance_score > rep_abundance:
55                              unique_structures[i] = (rep_structure, rep_file,
                                ↪  abundance_score, primitive_structure, rep_volume, rep_count
                                ↪  + 1)
56                          else:
57                              unique_structures[i] = (rep_structure, rep_file, rep_abundance,
                                ↪  abundant_structure, rep_volume, rep_count + 1)
58                      break
59
60              # If unique, add to the list with a count of 1
61              if is_unique:
62                  unique_structures.append((primitive_structure, file_path, abundance_score,
                    ↪  primitive_structure, cell_volume, 1))
63
64              if count % 1000 == 0:
65                  print(f"Processed {count} files...")
66
67      # Sorting the structures that the most popular would appear first
68      sorted_structures = sorted(unique_structures, key=lambda x: x[5], reverse=True)
69
70      return sorted_structures
71
```

```
72
73
74          def compare_frameworks_multiple_folders(main_dir):
75          """
76          Processes all .cif files in a single subdirectory, keeping the representative
77          file with the largest volume for each unique framework and saving them in the same
            ↪ subdirectory.
78
79          Parameters:
80          main_dir(str): Path to the main directory to process.
81          """
82          unique_structures = []  # List to store tuples of (structure, file_path,
            ↪ cell_volume, count)
83
84          # number of files processed
85          count = 0
86
87          # number of matching attempts
88          interactions = 0
89
90          # Initialize the structure matcher
91          matcher = StructureMatcher(comparator=FrameworkComparator())
92
93          for subdir in os.listdir(main_dir):
94          subdir_path = os.path.join(main_dir, subdir)
95          if os.path.isdir(subdir_path):
96          # Loop over each .cif file in the subdirectory
97          for file in os.listdir(subdir_path):
98          if file.endswith(".cif"):
99          count += 1
100         file_path = os.path.join(subdir_path, file)
101         # Load the structure from the CIF file
102         structure = Structure.from_file(file_path)
103         # Get the primitive unit cell
104         primitive_structure = structure.get_primitive_structure()
105         cell_volume = primitive_structure.volume
106
107         # Check if this structure matches any in the unique structures list
108         is_unique = True
109         for i, (rep_structure, rep_file, rep_volume, rep_count) in
            ↪ enumerate(unique_structures):
110         interactions += 1
111         if matcher.fit(primitive_structure, rep_structure):
112         is_unique = False
113         # Update the representative if the current structure has a larger volume
114         if cell_volume > rep_volume:
115         unique_structures[i] = (primitive_structure, file_path, cell_volume, rep_count + 1)
116         else:
117         # Increment the count for this framework
118         unique_structures[i] = (rep_structure, rep_file, rep_volume, rep_count + 1)
119         break
120
```

```
        # If unique, add to the list with a count of 1
        if is_unique:
        unique_structures.append((structure, file_path, cell_volume, 1))

        if count % 1000 == 0:
        print(f"Processed {count} files...")

        # Sorting the structures that the most popular would appear first
        sorted_structures = sorted(unique_structures, key=lambda x: x[3], reverse=True)

        print(f"Number of interactions: {interactions}")

        return sorted_structures
```

Listing 1: Core routine used to group structures into unique frameworks.

A.1.2 anx-notation-sort.py

Listing 2

```
        from pymatgen.core import Structure
        from pymatgen.core.composition import Composition
        import re
        import os
        from pymatgen.core.periodic_table import Element


        def formalize(comp, anion_list, max_ion_amount):
        notation = ''
        form_result = []
        if "(" in comp:
        comp = expand_formula(comp)
        comp_pattern = re.findall(r'(?P<element>[A-Z][a-z]*)(?P<count>\d*)', comp)
        counter = 0
        for element, num in comp_pattern:
        if element not in anion_list:
        if len(form_result) == 1 and form_result[0][0] == 'X':
        form_result.append(((chr(65)), num))
        counter = 1
        else:
        form_result.append(((chr(65+counter)), num))
        counter += 1
        else:
        if len(form_result) > 0 and form_result[-1][0] <= 'M':
        counter = 0
        for i in range(len(form_result)-1):
        if "X" in form_result[i][0]:
        counter = 1
        if counter < 3:
        form_result.append(((chr(88 + counter)), num))
        counter += 1
        else:
```

```
        form_result.append(((chr(80 + counter)), num))
        counter += 1
        form_result_sorted = sorted(form_result, key=lambda x: x[0])
        form_result_sorted = rearrange_cations(rearrange_anions(form_result_sorted))
        for item in form_result_sorted:
        if item[1]:
        if int(item[1]) > max_ion_amount:
        return None
        notation += item[0]
        notation += item[1]
        return notation


        def expand_formula(comp):
        # Pattern to match elements with optional lowercase letters and counts, inside and
        ↪  outside parentheses
        pattern =
        ↪  r'(\((?P<group>([A-Z][a-z]?\d*)+)\)(?P<multiplier>\d+)|(?P<element>[A-Z][a-z]*)(?P<count>\d*)?)'
        expanded_string = ""

        matches = re.finditer(pattern, comp)

        for match in matches:
        if match.group("group") and match.group("multiplier"):  # Case: Parenthesized group
        ↪  with multiplier
        group = match.group("group")
        multiplier = int(match.group("multiplier"))

        # Expand the elements inside the group
        inner_matches = re.findall(r'(?P<element>[A-Z][a-z]*)(?P<count>\d*)', group)
        for element, count in inner_matches:
        count = int(count) if count else 1
        expanded_count = count * multiplier
        expanded_string += f"{element}{expanded_count if expanded_count > 1 else ''}"

        elif match.group("element"):  # Case: Single element with optional count
        element = match.group("element")
        count = match.group("count")
        count = int(count) if count else 1
        expanded_string += f"{element}{count if count > 1 else ''}"

        return expanded_string


        def rearrange_cations(ls):
        count = 0
        for i in range(len(ls)):
        if ls[i][0] <= 'M':
        count = i

        ls[0:count + 1] = sorted(ls[0:count + 1], key=lambda x: int(x[1]) if x[1] else 1)
        for i in range(len(ls)):
```

```
        ls[i] = (chr(65 + i), ls[i][1]) if ls[i][0] < 'M' else ls[i]
        return ls


        def rearrange_anions(ls):
        count = 0
        for i in range(len(ls)):
        if ls[i][0] <= 'M':
        count = i

        ls[count+1: len(ls)] = sorted(ls[count+1: len(ls)], key=lambda x: int(x[1]) if x[1]
        ↪  else 1)
        for i in range(len(ls)):
        if i - count <= 3:
        ls[i] = (chr(87 + i - count), ls[i][1]) if ls[i][0] > 'M' else ls[i]
        else:
        ls[i] = (chr(79 + i - count), ls[i][1]) if ls[i][0] > 'M' else ls[i]
        return ls


        def is_metallic(composition):
        """
        Check if all elements in the composition are metallic.
        """
        for element in composition.elements:
        if not Element(element).is_metal:
        return False  # If any element is non-metallic, return False
        return True  # Return True only if all elements are metallic


        def is_single_element(composition):
        """
        Check if the material contains only a single element.
        """
        return len(composition.elements) == 1  # Return True if only one element is present


        def contains_hydrogen(composition):
        for element in composition.elements:
        if element == Element('H'):
        return True
        return False


        def contains_noble_gas(composition):
        return composition.contains_element_type("noble_gas")


        def sort_files(structure_directory, output_dir, anion_list, max_ion_amount,
        ↪  max_atoms):
        structure_files = [f for f in os.listdir(structure_directory) if
        ↪  f.endswith('.cif')]
```

```
        single_count = 0
        metal_count = 0
        hydrogen_count = 0
        noble_gas_count = 0
        long_structure_count = 0
        max_ion_count = 0
        lots_count = 0
        ionic_count = 0

        for file in structure_files:
        file_path = os.path.join(structure_directory, file)
        filename = os.path.basename(file_path)
        try:
        # Load the structure
        structure = Structure.from_file(file_path)

        # Skip if it contains hydrogen
        if contains_hydrogen(structure.composition):
        hydrogen_count += 1
        continue

        # Skip if it contains a noble gas
        if contains_noble_gas(structure.composition):
        noble_gas_count += 1
        continue

        # Skip if more than max_atoms in a unit cell
        if len(structure) > max_atoms:
        long_structure_count += 1
        continue

        formula = structure.composition.formula
        comp = Composition(formula)
        chem_not = comp.reduced_formula
        anx_notation = formalize(chem_not, anion_list, max_ion_amount)
        path = output_dir

        # Skip if there is more than max_ion_amount
        if not anx_notation:
        max_ion_count += 1
        continue

        # Skip if it does not contain either anions or cations
        if "A" not in anx_notation or "X" not in anx_notation:
        ionic_count += 1
        continue

        # Skip if it contains 4 or more cation/anions
        if "S" in anx_notation or "D" in anx_notation:
        lots_count += 1
        continue
```

```
183
184         if anx_notation:
185         with open(f"{path}/{anx_notation}.txt", "a") as f:
186         f.write(filename + "\n")
187
188         except Exception as e:
189         print(f"Failed to load {file_path}: {e}")
190         return None
191         print("Files sorted into lists")
192         print(f"Amount of structures filtered: \n"
193         f"Single atoms: {single_count}\n"
194         f"Alloys & intermetallics: {metal_count}\n"
195         f"contains hydrogen atoms: {hydrogen_count}\n"
196         f"contains noble gas atoms: {noble_gas_count}\n"
197         f"More than {max_atoms} atoms in unit cell: {long_structure_count}\n"
198         f"More than {max_ion_amount} of one species in formula unit: {max_ion_count}\n"
199         f"More than 3 different cations or anions species: {lots_count}\n"
200         f"Metarial contains only cations or anoins: {ionic_count}\n"
201         f"Total number of filtered structures: {single_count + metal_count + hydrogen_count
            ↪ + noble_gas_count + long_structure_count + max_ion_count + lots_count +
            ↪ ionic_count}\n")
202
203
204         if __name__ == '__main__':
205         structure_dir = './allgsexpstructures'
206         output_dir = './filter-count-test'
207         if not os.path.exists(output_dir):
208         os.makedirs(output_dir)
209         with open('anion_group1.txt', 'r') as f:
210         anion_group = [line.strip() for line in f.readlines()]
211         max_ion_amount = 4
212         max_atoms = 100
213         sort_files(structure_dir, output_dir, anion_group, max_ion_amount, max_atoms)
```

Listing 2: Core routine used to group structure files according to their stoichiometry.

תקציר

מבנה גבישי מציע בסיס קומפקטי, בלתי תלוי בהרכב, להבנת התנהגות חומרים ולהאצת גילוי חומרים. תזה זו מציגה תהליך עבודה אוטומטי וניתן לשחזור לזיהוי אבות-טיפוס מבניים ישירות מן הגאומטריות האטומיות, ולא מתוך מטא-נתונים של מסדי נתונים. המחקר מנתח 23,160 חומרים אנאורגניים במצב היסוד התרמודינמי שנצפו ניסויית מתוך Materials Project ומשווה מבנים על-ידי StructureMatcher של ספריית pymatgen באמצעות FrameworkComparator, כך שהתאמות נקבעות על סמך מסגרות גאומטריות ללא תלות בסוגי היסודות. ניתוח רגישות של טולרנסים משמש לקביעת ערכי הסף להתאמה (טולרנס אורכי 0.2, טולרנס זוויתי 5°, טולרנס מיקום 0.3), תוך איזון בין התאמת יתר וחסר ובמקביל שמירה על ישימות חישובית. יישום אלגוריתם לזיהוי פרוטוטיפים על כלל מערך החומרים במצב היסוד מניב 6,898 מסגרות מבניות ייחודיות, שמתוכן 2,410 משותפות לשני חומרים או יותר וזוהו כאבות-טיפוס מבניים. לאחר סינון נוסף וחלוקה מונחית סטויכיומטריה, זוהו 2,073 מסגרות מבניות ייחודיות ו-697 אבות-טיפוס מבניים. התוצאות מראות כי הפיזור של מבני של חומרים אנאורגניים מרוכז מאוד, כאשר מספר קטן של אבות-טיפוס מבניים אחראי לחלק גדול באופן בלתי פרופורציונלי מן החומרים היציבים המוכרים, וכי הטבע מעדיף מערכות בעלות סימטריה גבוהה בשל השונות הגאומטרית הגדולה יותר של סריגים בעלי סימטריה נמוכה תחת התאמה מחמירה.

אוניברסיטת תל - אביב

הפקולטה להנדסה ע״ש איבי ואלדר פליישמן
בית הספר לתארים מתקדמים ע״ש זנדמן-סליינר

אבות-הטיפוס המבניים של כל החומרים

חיבור זה הוגש כעבודת גמר לקראת התואר
״מוסמך אוניברסיטה״ במדע והנדסה של חומרים

על - ידי

רועי אשר

העבודה נעשתה במחלקה למדע והנדסה של חומרים
בהנחיית ד״ר לי ברטון

כסלו תשפ״ה