Machine Learning for Chemical Elements and Crystal Structures

Minoru Kusaba

Department of Statistical Science,

The Graduate University for Advanced Studies, SOKENDAI

Preface

Traditionally, synthesis of new materials and the investigation of their properties has been subject to the experience and intuition of individual researchers. However, in recent years, especially after the launch the Materials Genome Initiative (MGI) in the United States 2011, a new field of research called "materials informatics" has been receiving a lot of attention. Materials informatics is an emerging new interdisciplinary field of materials science and data science. The major focus of materials informatics is to develop a computational scheme that aims to improve efficiency in the development of new innovative materials.

In order to build the computational scheme, we need to solve the inverse problem, which refers to the task of predicting candidate materials with a given set of desired properties by finding the inverse map of the forward prediction model. Here, the forward predictive model refers to a statistical model that forwardly predicts the physicochemical properties of a given input material, and learning this model from a given material dataset can be considered the forward problem. In order to learn a statistical model from a material dataset, the input materials must be represented numerically in some way. The above three tasks can be regarded as "generation," "learning," and "representation" of material data. In this thesis, we present two studies in materials informatics on the representation and learning of material data, including chemical elements and crystal structures similarity.

In what follows, an overall overview of materials informatics is summarized in Chapter 1. With regard to the two studies mentioned above, an overview of materials informatics on inorganic materials is discussed in Chapter 2. Chapter 2 also clarifies the scientific contributions of the two studies in relation to the overall workflow. In Chapter 3, we present the study on automated design of periodic tables, formulating the task of periodic table design as a data visualization problem (or dimension reduction problem). In Chapter 4, we present the study on crystal structure prediction with machine learning-based element substitution, formulating the task of crystal structure prediction as a metric learning problem (the crystal structure prediction is a task to predict the stable or metastable crystalline state of a material with a given chemical composition). The main motivation of the studies is to show that the unique approaches constructed from the perspective of data science can provide new problem settings to the two fundamental problems in physical chemistry (the design of periodic table and the crystal structure prediction). Finally, Chapter 5 concludes this thesis.

Acknowledgment

I cannot find the right words to express my appreciation for my supervisor, Prof. Ryo Yoshida. I deeply appreciate his consistent support and suggestions. Without his supervision, I could not have progressed this far. I also thank him for his help in improving the written text.

I would like to thank Prof. Yukinori Koyama and Prof. Kiyoyuki Terakura for their advice and comments for the study on automated design of periodic tables. I also want to thank Dr. Chang Liu for his support and many helpful comments on both the studies in this thesis.

I am grateful to Hiromasa Tamaki, Tomoyasu Yokoyama, Kensuke Wakasugi, Koki Ueno, and Satoshi Yotsuhashi from Panasonic Corporation for their helpful discussions and their generosity in providing us with a list of benchmark crystals for the study on crystal structure prediction with machine learning-based element substitution.

I would like to thank Prof. Stephen Wu, Prof. Hideitsu Hino, Prof. Daichi Mochihashi, and Prof. Takashi Miyake for their constructive comments during the review process of this thesis.

During my PhD course, I have been supported and encouraged by my professors, colleagues, and friends—I would like to thank all of them.

Contents

1 Introduction

The rapid development of materials databases and recent advances in machine learning techniques have led to a significant shift in the field of materials science, giving rise to a new interdisciplinary field called "materials informatics." This field utilizes a combination of machine learning techniques and digital technologies to promote the development and discovery of new materials as well as to further our understanding of material systems [1]. The major focus of materials informatics is to develop a computational scheme that can improve efficiency in the development of new innovative materials (computational materials design) [2]. In general, the design space of materials research is considerably vast. For example, it is estimated that there are about 10^{60} candidate molecules in the chemical space of small organic molecules alone [3]. On the contrary, the number of synthesized molecules in public databases is in the order of 10^8 at most. Therefore, there is still a vast unexplored area in the chemical space. Furthermore, in the study of advanced materials, the size of the design space increases drastically with the addition of various design parameters such as the selection of additives and solvents, compositional features in the fabrication of composite materials, various kinds of processing conditions, and so on. Using machine learning as a technological driving force, new materials with innovative properties can be discovered from such vast search space. This is precisely the primary objective of this emerging new field.

The basic workflow of materials informatics consists of forward and inverse problems (Fig. 1.1). For example, the objective of the forward problem is to obtain a statistical model $Y = f(X)$ that forwardly predicts physicochemical properties Y of any given input material X. The inverse problem, on the contrary, predicts candidate materials X with a given set of desired properties $Y = Y^*$ by finding the inverse map of the forward model. The workflow is common and not worthy of special mention, but one of the distinctive features of data analysis in materials informatics lies in the particularity and high-dimensionality of the input variable X to be handled. Variables such as chemical compositions, molecules, crystal structures, etc. are generally non-trivial to represent numerically as fixed-length vectors. Therefore, in order to formulate such a scientific task within the framework of data science, we need to design descriptors that quantify the patterns of X. In addition, to solve the inverse problem, we need a generative model of X that can move freely in a vast search space.

The task can be seen as "representation," "learning," and "generation" of material data. An input variable such as chemical composition, molecule, or crystal structure is numerically "represented" into a descriptor vector, and the mapping from the vectorized input to an output property is "learned" with a given dataset. The inverse mapping of the model is then explored by computationally "generating" materials with desired properties to identify promising candidates. In this thesis, we will present two studies on the representation and learning of material data, including chemical elements and crystal structures similarity, in Chapters 3 and 4, respectively.

Data representation and visualization are important machine learning techniques that aim to computationally visualize high-dimensional data [4]. In particular, data visualization is helpful in understanding the overall picture of the distribution of high-dimensional materials data. In Chapter 3, we present the study on computational design of periodic tables using an unsupervised machine learning algorithm [5]. The object to be represented and visualized here is a set of chemical elements with their observed physicochemical properties. An excellent way of representing element species has already been well-established as the periodic table of the chemical elements; in 1869, the first draft of the periodic table was developed and published by Russian chemist Dmitri Mendeleev [6]. In terms of data science, his achievement can be viewed as a successful example of feature embedding based on human cognition; chemical properties of all known elements at that time were compressed onto the two-dimensional grid system for a tabular display. In this thesis, we seek to answer whether machine learning can reproduce or recreate the periodic table by using observed physicochemical properties of the elements. To achieve this goal, we developed a periodic table generator (PTG). The PTG is an unsupervised machine learning algorithm based on generative topographic mapping (GTM) [7] that can automate the translation of high-dimensional data into a tabular form with varying layouts. PTG autonomously has produced various arrangements of chemical symbols, which organize a two-dimensional array such as Mendeleev's periodic table or three-dimensional spiral table according to the underlying periodicity in the given data. We further show what PTG learns from the element data and how the element features, such as melting point and electronegativity, are compressed to the lower-dimensional latent spaces. The related literature and background of this study are summarized in Sections 2.7 and 2.8. The GTM, which is the basis of the proposed method, is described in detail in Sections 2.9 and 2.10.

In Chapter 4, we describe the study on data-driven crystal structure prediction using structural similarity. The goal of the crystal structure prediction is to predict and discover the stable or metastable crystalline state of a material with a given chemical composition. This prediction task is a critical and inevitable part in the entire workflow of the computational design of inorganic solid-state materials. The problem of predicting the crystal structure formed by an arbitrary chemical composition remains unsolved in solid-state physics. In principle, stable or metastable structures formed by atomic or molecular assemblies can be found by solving a local optimization problem for a potential energy surface defined on the space of atomic coordinates. The major approach to the computational crystal structure prediction relies on the repeated calculation of first-principles potential energy surfaces, such as density functional theory (DFT) calculations [8, 9]. However, for complex systems, such as the one with many atoms per unit cell, the iterative gradient descent on the potential energy surface using first-principles calculations is prohibitively expensive. It is known that such an ab initio approach cannot identify crystals with more than 30-40 atoms per unit cell. In this thesis, we present a robust method for crystal structure prediction based on element substitution using machine-

learned crystal structure similarity. The method relies on a machine learning algorithm referred to as metric learning [10]. This algorithm is used to learn the representation of crystal structure similarity with a given dataset and to automate the selection of template structures from a crystal structure database that have high chemical replaceability to the unknown stable structure for a given chemical composition. In metric learning, a binary classifier is constructed to determine whether the crystal structures of two given chemical compositions are identical or not. Here, crystals with sufficiently high structural similarity are treated as identical, and the labeled dataset is extracted from the crystal structure database. A broader background of this study and literature related to the computational design of inorganic solid-state materials is included in Sections 2.1 to 2.6.

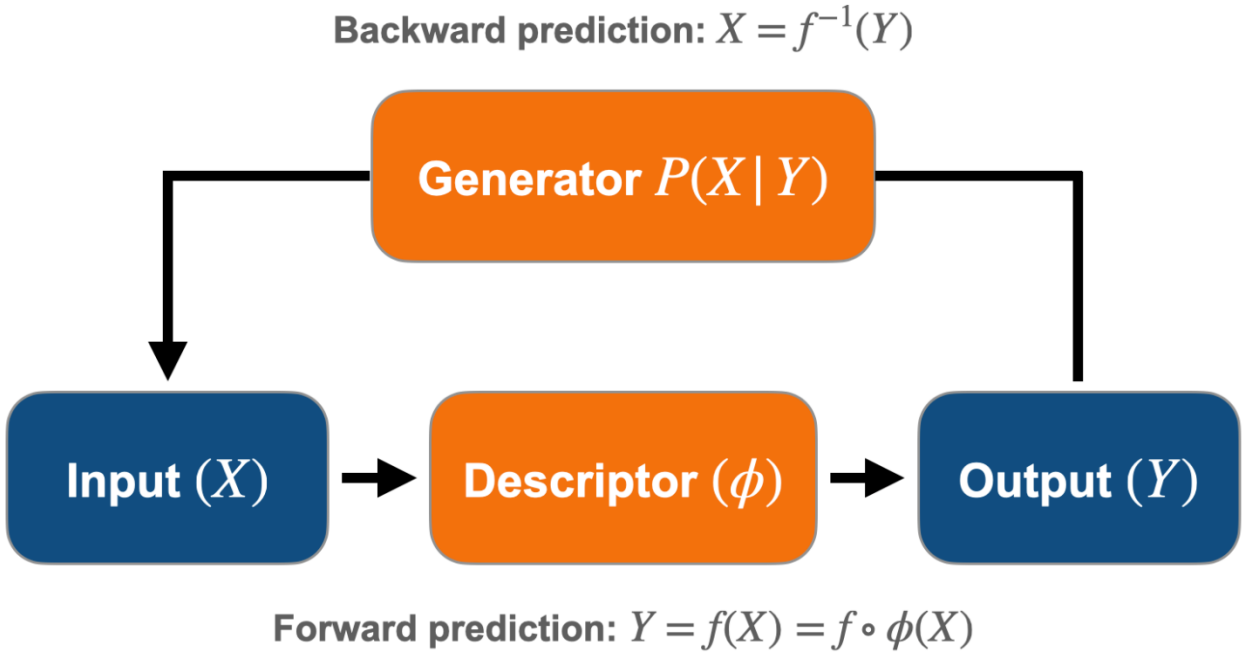


Figure 1.1. Basic workflow of materials informatics

2 Review of materials informatics on inorganic materials

Here, we briefly discuss the history and recent developments in materials informatics on inorganic materials. In particular, we aim to clarify the scientific contribution of our two studies, "recreation of the periodic table using an unsupervised machine learning algorithm" and "crystal structure prediction based on machine learning representation of crystal structure similarity," in relation to the overall workflow. In Sections 2.1 to 2.6, an overview and recent studies related to the computational design of inorganic solid-state materials are discussed. Particularly, crystal structure prediction is discussed in Section 2.6 and s representation learning for materials data including chemical element species is discussed in Sections 2.7 and 2.8. Since the periodic table generation described in Chapter 3 relies on the GTM and its variant called GTM with latent variable dependent length-scale and variance (GTM-LDLV) [11], their methodological basis is discussed in Sections 2.9 and 2.10, respectively.

2.1 Introduction

The field of materials informatics offers various machine learning techniques that enable us to improve the efficiency of development of new materials found via data-driven method [1]. Traditionally, new inorganic compounds have been discovered through trial-and-error based on human intuition and through laborious, time-consuming synthetic experiments. However, thanks to advances in ab initio first-principles calculations, such as density functional theory (DFT) calculations [8, 9], materials research based on theory and computational science has played a major role in the past decades. High accuracy and increasing efficiency of the DFT calculations has enabled researchers to perform comprehensive studies on a large number of compounds, leading to the rapid expansion of DFT-based materials properties and crystal structure databases (as detailed in Section 2.2). Furthermore, the accumulation of massive data has greatly facilitated a wide variety of successful applications of machine learning. In recent years, more advanced methods, which combine accurate but time-consuming DFT calculations and fast machine learning techniques, have been proposed.

The basic workflow of the computational materials design is illustrated in Fig. 2.1. The individual tasks (such as "representation of materials") are detailed in Sections 2.2 to 2.6. In materials informatics of inorganic crystalline materials, the most fundamental input variable is the chemical composition, which represents the content of element species consisting of a material. Given the chemical composition and the temperature and pressure of the system, the crystal structure of assembled atoms in a stable or metastable state can, in principle, be determined ab initio. The crystal structure can be predicted by iteratively evaluating the potential energy function defined on the atomic coordinates using first-principles calculations to find the local minimum solution. While the problem formulation is clear, due to the high computational cost of iterative first-principles calculations, such ab initio approaches are still unable to solve crystalline systems with a large number of atoms, such as 30-40 atoms per unit cell. In Chapter 4, we describe an ultra-efficient and accurate workflow for crystal structure prediction that utilizes a metric learning technique that learns the relationship between the similarity of chemical composition and the one of crystal structures from existing crystal data. The method does not require any first-principles calculations, except for a validation step.

Once a crystal structure is determined, the mapping to physical properties can be obtained by first-principles calculations, which again require significant computational resources. In summary, forward prediction consists of two steps: from chemical composition to crystal structure, and from crystal structure to properties. Recently, various attempts have been made to speed-up the computation of these two tasks by using surrogate machine learning models. The task of predicting properties from crystal structures using machine learning has been a central problem in materials informatics from its early days. There has also been much work on predicting properties directly from chemical composition, ignoring the crystal structure in the middle of the workflow. The key issue here is the design of descriptors for the chemical composition and crystal structure. The chemical composition consists of a list of elements and their contents. The number of elements, the constituent units, varies across different materials. For example, the number of elemental species differs between binary and ternary compounds. In order to deal with such set variables, a vector representation is usually defined by combining a predefined set of elemental features with compositional ratios and taking their summary statistics such as weighted average and weighted variance (Section 2.3). In the study on the periodic table, discussed in Chapter 3, unsupervised learning was applied to obtain feature representations of elements from data on physicochemical properties. For the representation of crystal structures, descriptors that numerically describe the topology, geometry, size, symmetry, etc. of the atomic arrangement were used, with the feature units representing the surrounding environment of each atom (Section 2.3). Another approach is to include calculated or experimental values of physical properties such as lattice parameter, band gap, density of states, etc. in the descriptors. However, descriptors including physical properties are computationally very expensive, and the resulting predictive models are not suitable for exhaustive screening. Nonetheless, once such a vector representation is obtained, the mapping from vectorized materials to real-valued properties can be estimated by applying conventional supervised learning.

Once a forward prediction model $Y = f(X)$ is obtained, its inverse map is obtained to predict a material X with an arbitrary property $Y = Y^*$. The simplest way to solve the inverse problem is virtual high-throughput screening. A library of a large number of candidate inputs is constructed and then screening experiments are carried out using the trained model. In general, the computational cost of machine learning models is much lower than that of experiments or theoretical calculations, and thus a large number of candidate materials can be evaluated. Although machine learning has been used to screen materials for drug discovery since a long time, it has only recently been applied to materials research. Gómez-Bombarelli et al. [3] used

a neural network trained on first-principles data to screen more than 400,000 candidate materials and discover new molecules for organic LEDs with high external quantum yields. Seko et al. [12] calculated the lattice thermal conductivity of 101 inorganic compounds using first-principles calculations, and combined Bayesian optimization [13] and Gaussian process regression [14] to derive a property prediction model. Using this model, they screened 54,779 compounds in the Materials Project [15] and identified 221 compounds with low thermal conductivity. Carrete et al. [16] used the theoretical thermal conductivity values of 32 half-Heusler compounds to derive a regression model using random forest algorithm and 450 low thermal conductivity compounds registered in the AFLOW database were screened. Pilania et al. [17] calculated 8 physical properties (band gap, formation energy, dielectric constant, etc.) for 175 polymeric materials (polyethylene) whose repeating units consisted of four blocks of basic elements using first-principles calculations and then applied kernel ridge regression. The prediction model of each property was constructed by applying kernel ridge regression. Using this model, 29,365 polymer materials with 8 blocks of polymer units were screened. Wu et al. [18] synthesized a new polymer with high thermal conductivity by deriving a model to predict thermophysical properties using the polymer property database PoLyInfo [19]. A virtual library was created using a Bayesian molecular generation algorithm, and the three polymers predicted to have high thermal conductivity were selected for experimental validation. To derive a predictive model of thermal conductivity from a small amount of data, transition learning was utilized; the prediction model was derived from other physical properties of polymers that are correlated with thermal conductivity, such as glass transition temperature and specific heat, and the pretrained model was fine-tuned using a small amount of data to obtain a highly accurate prediction model of thermal conductivity. In Chapter 4, we construct a binary classification model that predicts the crystal structure identity of any two chemical compositions and we also perform a virtual screening to predict stable crystal structures. Specifically, a query composition with an unknown crystal structure and a chemical composition with a known crystal structure is input to the model, and the chemical composition with the same structure is identified.

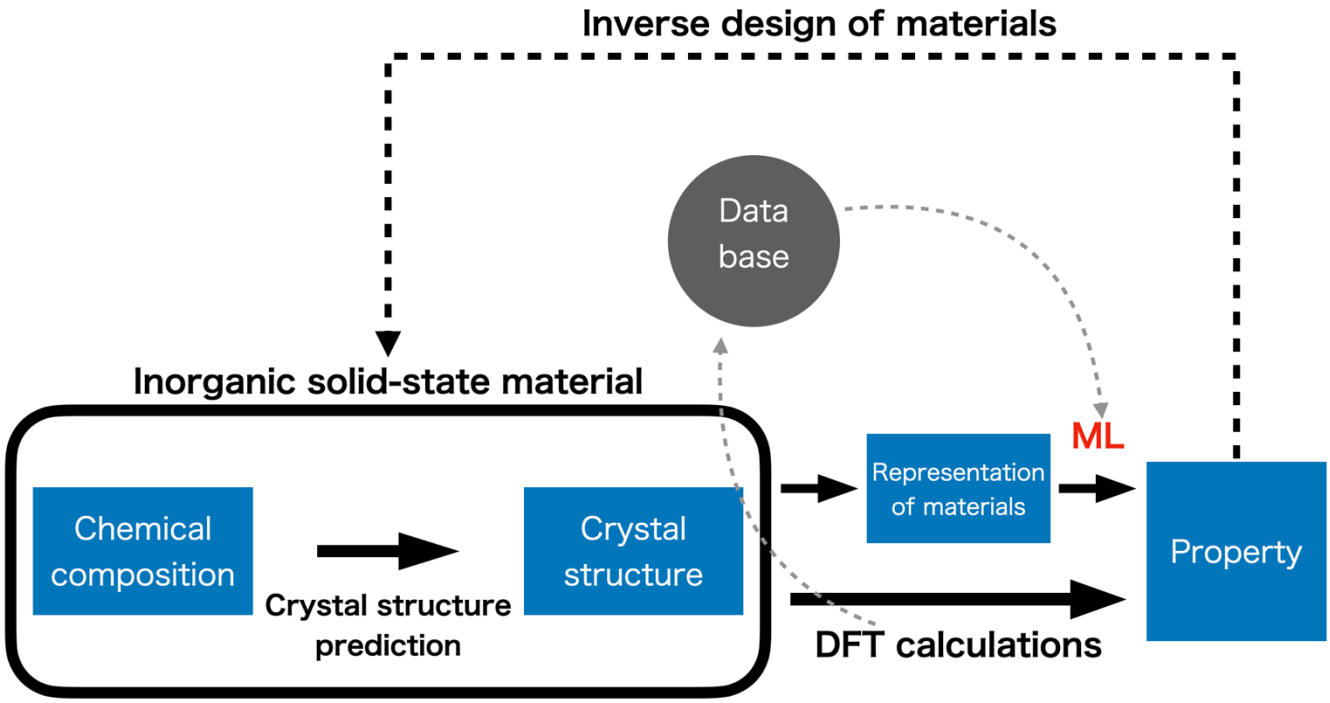


Figure 2.1. Basic workflow of the computational design of inorganic solid-state materials

2.2 Crystal structure databases

The emergence of comprehensive databases of crystal structures has played a key role in the development and widespread use of materials informatics. Various databases have been created so far, such as the Cambridge Crystallographic Data Centre (CCDC) in 1965 [20], the Inorganic Crystal Structure Database (ICSD) in 1983 [21], CRYSTMET in 1993 [22], Pauling File in 2002 [23], the Crystallography Open Database (COD) in 2003 [24], and Pearson's Crystal Data (PCD) in 2007 [25]. Furthermore, thanks to DFT calculations and increasing computing power, a wide variety of computational materials databases, such as the Predicted Crystallography Open Database (PCOD) in 2005, the Materials Project (MP) database [15,26] and the Automatic FLOW for Materials Discovery database (AFLOW) in 2011 [27], the Harvard Clean Energy Project (CEP) [28] and the Open Quantum Materials Database (OQMD) in 2013 [29], and the Novel Materials Discovery Laboratory (NOMAD) in 2015 [30], have been created. The Materials Project database was used for both the studies described in Chapters 3 and 4. While

the development of such databases has enhanced the potential of machine learning approaches in materials science, in order to apply them to machine learning, it is necessary to transform the data into a form (a descriptor vector) that is readable by machine learning algorithms. In the next section, we provide an overview of existing representation techniques for crystal structure information.

2.3 Representation of inorganic materials

Compositional descriptors express the amounts of chemical elements. For example, a conventional compositional descriptor operates with a predefined set of element features, such as electronegativity and atomic weight [31, 32]. For a given composition ratio, the feature values of constituent elements are collapsed to a quantity to describe a compositional feature; for example, by using the weighted mean and weighted variance of the element features. Here, we describe the compositional descriptor. The chemical formula is denoted by $X = X_{c^1}^1 X_{c^2}^2 \cdots X_{c^K}^K$. Each element of the descriptor vector takes the following form:

$$\phi_{g,\eta}(X) = g\big(c^1, \dots, c^K, \eta(X^1), \dots, \eta(X^K)\big).$$

The notation $\eta(X^K)$ on the right-hand side denotes a feature quantity of element X^K, such as the atomic weight, electronegativity, or polarizability. With the function g, the K element features $\eta(X^1), \dots, \eta(X^K)$ with fraction $c^1, \dots, c^K$ are converted into the compositional feature. For g, we can operate with the weighted average, weighted variance, max-pooling, and min-pooling, as given by

$$\phi_{\mathrm{ave},\eta}(X) = \frac{1}{\sum_{k=1}^K c^K} \sum_{k=1}^K c^K \eta(X^K),$$

$$\phi_{\mathrm{var},\eta}(X) = \frac{1}{\sum_{k=1}^K c^K} \sum_{k=1}^K c^K \left(\eta(X^K) - \phi_{\mathrm{ave},\eta}(X)\right)^2,$$

$$\phi_{\max,\eta}(X) = \max\{\eta(X^1), \dots, \eta(X^K)\},$$

$$\phi_{\min,\eta}(X) = \min\{\eta(X^1), \dots, \eta(X^K)\}.$$

Table. 4.2 provides a list of the 58 element features used in the analysis in Chapter 4 that were implemented in XenonPy, a Python open-source platform for materials informatics [33]. The element feature set includes the atomic number, bond radius, van der Waals radius, electronegativity, thermal conductivity, bandgap, polarizability, boiling point, melting point, number of valence electrons in each orbital, and so on.

Due to the complexity of 3D periodic crystal structures of inorganic solid-state materials, the task to encode the structural information into a finite-length descriptor vector is not obvious [2]. Therefore, the representation of inorganic material information itself is an important research subject in this field, and various representation methods have been proposed so far [34]. In some studies, only indirect information (ignoring complex information like crystal structure) such as chemical composition is used as a descriptor [35, 36]. However, it is reported that the inclusion of structural information such as radial distribution function could significantly improve the prediction accuracy even further (Seko et al. [31]).

The properties of ideal representation should include invariance, uniqueness, stability, and interpretability [34]. As a representation method of crystal structure, Zimmermann and Jain proposed local structure order parameters [37]. This method calculates a vector-type descriptor (site fingerprint) for each atomic site in the crystal structure by evaluating the degree of resemblance of the coordination environment of an atomic site to the preset-coordination motifs. Then, a crystal structure descriptor is calculated by taking statistics for each element of the site fingerprints across all atomic sites in the crystal structure. Most representation methods of crystal structure based on local environment of each atom calculate a crystal structure descriptor similar to this procedure. Bartok et al. [38] proposed a representation method based on smooth overlap of atomic positions (SOAP). Rupp et al. [39] proposed the Coulomb matrix, which represents nuclear coulombic interaction. Pham et al. [40] proposed the orbital field matrix, a representation based on the valence shell electrons of neighboring atoms. Schutt et al. proposed a novel crystal structure representation based on the averaged partial radial distribution function of pairwise distances between atoms. Representation methods based on connectivity constitute another category of effective methods. Application of N-grams, which are histograms of unique coordination environments and edge sequences, as a descriptor has been reported as being effective in predicting formation energies and electronic band gaps [41]. Furthermore, Xie et al. [42] proposed a crystal graph convolutional neural network framework to directly learn material properties from the connection of atoms in the crystal, providing a universal and interpretable representation of crystalline materials. In some recent studies [43, 44], 3D image-based representations of crystal structure have been used for learning of generative neural network-based models such as variational autoencoder (VAE) [45] and generative adversarial network (GAN) [46]. For example, Noh et al. [43] proposed the first generative model (VAE-based method) for inorganic solid-state materials using a 3D atomic image representation. In this study, complex 3D atomic image representations were encoded to the highly simplified continuous latent space with VAE, enabling optimization of materials on the latent space. Various representation methods are implemented in Python libraries for materials analysis such as pymatgen [47] and matminer [48, 49].

2.4 Property prediction

Property prediction involves the prediction of properties of materials using compositional features or structures of materials as input. This forward mapping with machine learning corresponds to the forward arrow, denoted as "ML (Machine Learning)" in Fig. 2.1. Such an ML-aided method provides fast prediction of materials, enabling computational screening of large-scale materials data. ML-aided property prediction is achieved by combining representation methods (as described in previous section) and supervised ML models such as kernel ridge regression, random forest, boosting methods, and neural network frameworks [50].

2.5 Inverse design strategies

Inverse design of materials is defined as the task to search and output materials with a predefined set of target properties. This inverse mapping task corresponds to the reverse arrow ("inverse design of materials") in Fig. 2.1. The inverse design strategies can be categorized into three types [2]: (1) high-throughput virtual screening (HTVS), (2) global optimization (GO), and (3) generative ML models.

HTVS screens a predefined set of candidate materials using a property evaluation scheme. For property evaluation, DFT calculations are often used. However, since the computational cost of DFT calculations is considerably high, ML surrogate models have been applied for exploring a large materials space. Finally, the candidate materials proposed by such computational workflows are verified in real experiments.

The inverse design of materials with GO aims to find the target materials by optimizing a forwardly predicted property surface with respect to input materials with an optimization algorithm such as the evolutionary algorithm. The property surface to be optimized is evaluated using first-principles calculations or ML models. For example, Podryabinkin et al. [51] proposed a crystal structure prediction algorithm that is based on the evolutionary algorithm and the machine learning interatomic potentials actively learning on-the-fly, which is implemented using the USPEX software [52]. The method aims to find the stable crystal structure by optimizing ML-predicted interatomic potentials using the evolutionary algorithm. The ML models for interatomic potentials are actively learned on-the-fly with the aid of DFT calculations for adaptive data acquisition.

The inverse design of materials with generative ML models is another promising strategy. The generative ML models have been utilized to generate virtual materials data from the lower-dimensional continuous latent space learned from the prior knowledge on materials data distribution [53]. This strategy has the potential to generate new materials with target properties not present in existing databases. Furthermore, materials representation in the lower-dimensional continuous latent space largely facilitates exploration of materials. For example, Hoffmann et al. [44] proposed a general-purpose encoding-decoding framework for 3D atomic density under VAE formalism. The study by Noh et al. [43] is another example of using generative ML models, which was briefly discussed in Section 2.3.

2.6 Crystal structure prediction

In the entire workflow of discovering new materials, the process of predicting the stable crystal structure of a particular chemical compound (crystal structure prediction; CSP) is a critical and inevitable part, as shown in Fig. 2.1. In recent years, various methods for computational CSP have been proposed. The major approach to computational CSP relies on the repeated calculation of first-principles potential energy surfaces, such as DFT calculations [8, 9]. As optimization methods, random search [54, 55, 56], simulated annealing [57, 58], Basin-hopping [59], minima hopping [60, 61], evolutionary algorithm (EA) [52, 62, 63], particle-swarm optimization (PSO) [64, 65], Bayesian optimization (BO) [66], and look ahead based on quadratic approximation (LAQA) [67] have been applied so far. More recently, as a promising alternative, machine learning interatomic potentials have attracted considerable attention for substantially speeding-up the optimization process by bypassing the time-consuming ab initio calculations [68, 69]. These existing methods can be classified into two types: sequential search and batch selection. The former set of methods, such as EA and PSO, explores the global or local minimum of the potential energy surface by iteratively modifying a current set of candidate crystalline forms with a predefined set of genetic manipulations in which ab initio structural optimization is repeatedly applied to the currently obtained candidates. The batch selection methods, such as BO and LAQA, utilize surrogate models learned with a training set of DFT energies and crystal structures for narrowing down to more promising candidates with lower predicted energies from a predefined set of candidate crystals. In both cases, a reasonable set of initial structures needs to be created by using a crystal structure generator. In this regard, the random symmetric structure generator [52, 70] and the topology-based structure generator [71] have been proposed so far. Nonetheless, the existing methods rely on the iterative use of computationally expensive ab initio energy calculations.

Another type of computational CSP is based on element substitution [72, 73]. Historically, most crystals synthesized so far have been discovered by element substitution of previously discovered ones. Substitution-based CSP mimics such traditional protocols computationally; it aims to predict the stable crystal structure by replacing elements in an already known template crystal that possesses high chemical replaceability to the target structure to be predicted. The chemical replaceability can be statistically estimated by learning the co-occurrence pattern of element pairs in a crystal structure database [73]. Such substitution-based methods do not require time-consuming potential energy calculations except in the process of locally optimizing replaced crystals. However, unlike the ab initio energy-based methods, the template-based methods lack the ability to predict completely new crystal structures. However, in spite of this limitation, the template-based methods are far less computationally expensive than the ab initio energy-based methods, and are known to be sufficiently useful for the prediction

of many crystal structures [72].

In Chapter 4, we propose a new CSP framework that conducts the prediction task by selecting crystal structures that are predicted to be similar to the stable structure of a given chemical composition from the existing crystal structures in a database. The method is classified under element substitution-based methods, in which the replaceability of two chemical elements is statistically estimated based on the observed frequency of their occurrence in two similar crystal structures. The great potential of the present method is demonstrated through the prediction of a wide variety of crystalline systems in Chapter 4.

2.7 Dimension reduction and visualization of materials data

Visualization of high-dimensional materials data is very important in various aspects of materials informatics. The data visualization methods differ depending on whether a material is represented in a graph or vector. For the graph-based descriptor, visualization methods such as the scaffold tree [74] or scaffold network [75] are often used. For the vector-based descriptor, various dimensionality reduction techniques [4] are available, such as principal component analysis (PCA), kernel PCA [76], isometric feature mapping (ISOMAP) [77], local linear embedding (LLE) [78], and t-distributed stochastic neighbor embedding (t-SNE) [79].

Among such various methods for dimensionality reduction techniques, PCA is the most popular, traditional method, widely used for the visualization of materials data. For example, Cender et al. [80] used PCA to data mine missing information in ab initio libraries of alloys versus structure prototypes. Suh and Rajan [81] used PCA to show that structure maps representing structure–property relationships (electronic features and crystal structure parameters) can be reproduced via data mining. As a nonlinear dimensionality reduction technique, t-SNE is frequently used for visualizing materials data. For example, Zhong et al. [82] used t-SNE to show the distribution of the CO adsorption energies over adsorption sites on Cu-containing alloys with different local atomic environments and compositional features. Using t-SNE, they also revealed that Cu-Al exhibits the highest abundance of adsorption sites and site types with near-optimal CO adsorption energy values.

When the number of data is large, some dimensionality reduction methods become difficult to apply due to computational cost constraints [83]. For instance, multidimensional scaling [84] and t-SNE do not scale with the amount of data because of their high memory requirements. To visualize large databases onto a low-dimensional space, PCA is widely used. For example, Singh et al. [85], Le Guilloux et al. [86], and Reymond [87], used PCA for large-scale chemical space analysis. While PCA is a powerful method even for large data, if the data is distributed in a lower-dimensional nonlinear manifold, its features may not be fully captured with PCA since it takes into account only on linear projections. As a nonlinear dimensionality reduction technique that is applicable to large data, self-organizing map (SOM) [88] is a widely used effective method. For example, Horvath et al. [89] used SOM to analyze a large database of approximately 200,000 molecules. SOM has the potential to generate more information-rich plots than PCA. However, SOM is a rule-based naive algorithm where the objective function to be optimized may not be defined, which hampers its application in certain cases [90]. Bishop et al. [7] developed generative topographic mapping (GTM) as a probabilistic extension of SOM, which strictly follows the framework of Bayesian inference. Gaspar et al. [83] successfully visualized large-scale chemical data with an incremental version of GTM [91]. They also performed a statistical analysis on this data, taking advantage of the fact that GTM is a well-defined generative model.

2.8 Visualization of chemical elements as a periodic table

Elements are the most important building blocks in physical chemistry. The representation and visualization of elemental features is a fundamental problem in physical chemistry that has already been studied for more than 150 years, long before the birth of modern data science. The present periodic table is considered as a product of data analysis from 150 years ago. The periodic table is a tabular arrangement of elements that is designed such that the periodic patterns of their physical and chemical properties are clearly understood. The prototype of the current periodic table was first presented by Mendeleev in 1869 [6]. At that time, approximately 60 elements (and a few of their chemical properties) were known. When the elements were arranged according to their atomic weight, Mendeleev noticed an apparent periodicity and an increasing regularity. Inspired by this discovery, he constructed the first periodic table. Despite the subsequent emergence of significant discoveries [92, 93], including the modern quantum mechanical theory of the atomic structure, Mendeleev's achievement is still the de facto standard. Regardless, the design of the periodic table continues to evolve, and hundreds of periodic tables have been proposed in the last 150 years [94, 95]. The structures of these proposed tables have not been limited to the two-dimensional tabular form, but have also included spiral, loop, and three-dimensional pyramid forms [96, 97, 98].

The periodic tables proposed so far have been products of human intelligence. However, a recent study has attempted to redesign the periodic table using computer intelligence, that is, machine learning [99]. Through this approach, building a periodic table can be viewed as an unsupervised learning task. Precisely, the observed physicochemical properties of elements are mapped onto regular grid points in a two-dimensional latent space such that the configured chemical symbols adequately capture the underlying periodicity and similarity of the elements. Lemes and Pino [99] used SOM [88] to place five-dimensional features of elements (i.e., atomic weight, radius of connection, atomic radius, melting point, and reaction with oxygen) into two-dimensional rectangular grids. This method successfully placed similarly behaving elements into neighboring sub-regions in the lower-dimensional spaces. Zhou et al. [100] suggested the machine-learned properties of atoms from the extensive database of known compounds and materials themselves. Although this study did not aim to recreate the periodic table directly, it is

noteworthy as an applied study of machine learning on information about chemical elements. The machine-learned properties suggested in this study are represented in terms of high-dimensional vectors, and their PCA projections showed clustering of atoms into meaningful groups consistent with human knowledge. However, the machine learning algorithms never reached Mendeleev's achievement as they missed important features such as between-group and between-family similarities.

In Chapter 3, in order to reproduce the visualization results as closely as possible to Mendeleev's achievements by machine learning, we propose a periodic table generator (PTG). The PTG is an unsupervised machine learning algorithm based on the GTM that can automate the translation of high-dimensional data into a tabular form with varying layouts on-demand. The PTG can autonomously produce various arrangements of chemical symbols, which organize a two-dimensional array such as Mendeleev's periodic table or three-dimensional spiral table according to the underlying periodicity in the given data. Since PTG can be considered as an extension of GTM, details regarding GTM are discussed in the next section.

2.9 GTM

GTM is a latent variable model that represents the probability density of data using a nonlinear function of lower-dimensional latent variables. It can be regarded as a probabilistic variant of SOM.

In GTM, K grid points (called "nodes" hereafter) $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_K$ regularly arranged in the L-dimensional latent space are prepared for data visualization, and consider a nonlinear function $\boldsymbol{f}(\boldsymbol{u}_k; \boldsymbol{\theta})$ that maps the nodes $\boldsymbol{u}_k$ to a point $\boldsymbol{y}_k$ on the D-dimensional feature space. The dimension of the latent space L is set to be less than 3 for visualization. $\boldsymbol{\theta}$ is a parameter set that determines $\boldsymbol{f}(\boldsymbol{u}_k; \boldsymbol{\theta})$. It is assumed that the D-dimensional feature vector $\boldsymbol{x}_n$ is generated independently by a restricted mixture of K Gaussian distributions, where all mixing coefficients are $1/K$, the mean of the Gaussian distribution is $\boldsymbol{y}_k$, and the covariance matrix is all $\beta^{-1}\boldsymbol{I}$. Then, the distribution is given by

$$p(\boldsymbol{x}_n|\boldsymbol{\theta}, \beta) = \frac{1}{K}\sum_{k=1}^{K} p(\boldsymbol{x}_n|\boldsymbol{u}_k, \boldsymbol{\theta}, \beta),$$

$$p(\boldsymbol{x}_n|\boldsymbol{u}_k, \boldsymbol{\theta}, \beta) = N(\boldsymbol{x}_n|\boldsymbol{y}_k, \beta^{-1}\boldsymbol{I}), \qquad \boldsymbol{y}_k = \boldsymbol{f}(\boldsymbol{u}_k; \boldsymbol{\theta}),$$

where $N(\cdot\,|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the Gaussian density function with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Here, we introduce a vector of K latent variables, $\boldsymbol{z}_n = (z_{1n}, \cdots, z_{Kn})'$. The kth entry z_{kn} takes the value 1 if $\boldsymbol{x}_n$ is generated by the kth component distribution, and 0 otherwise. Here, let $\boldsymbol{X}$ denote a matrix of $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N$ elements and $\boldsymbol{Z}$ be a matrix of $\boldsymbol{z}_1, \cdots, \boldsymbol{z}_N$. Then, their joint distribution is given by

$$p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta}, \beta) = K^{-N}\prod_{n=1}^{N}\prod_{k=1}^{K} N(\boldsymbol{x}_n|\boldsymbol{y}_k, \beta^{-1}\boldsymbol{I})^{z_{Kn}}. \tag{2.1}$$

If the function $\boldsymbol{f}(\boldsymbol{u}_k; \boldsymbol{\theta})$ is a smooth nonlinear function, then nodes $\boldsymbol{u}_k$ are mapped onto $\boldsymbol{y}_k$ while maintaining the topological relationship in the latent space. GTM is seen as a mixture of Gaussian distributions, which means $\boldsymbol{y}_k$ are restricted to the lower-dimensional manifold.

In GTM, the function $\boldsymbol{f}(\boldsymbol{u}_k; \boldsymbol{\theta})$ is constructed by a Gaussian process (GP). The nature of the GP is determined by the choice of covariance function. The conventional GTM model uses a covariance function with a constant length-scale throughout the latent space. This model cannot locally change the smoothness of the nonlinear function representing the distribution of the observed data according to the value of the latent variable. The underlying patterns of the element data are considered nonlinear and highly complex; thus, we require a GTM model that can represent more flexible functions. Therefore, we focused on GTM-LDLV [11], which is a recently proposed GTM model that can control the smoothness of the nonlinear function locally according to the value of the latent variable. Details regarding GTM-LDLV are described in the next section.

2.10 GTM-LDLV

In GTM-LDLV, it is assumed that the D-dimensional feature vector $\boldsymbol{x}_n$ is generated independently by a restricted mixture of K Gaussian distributions defined in equation (2.1), and the nonlinear function $\boldsymbol{f}(\boldsymbol{u}_k)$ is modeled to be the product of two functions: a D-dimensional vector-valued function $\boldsymbol{h}(\boldsymbol{u}_k)$ and a positive scalar function $g(\boldsymbol{u}_k)$. Then, their joint distribution is given by

$$p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{g}, \boldsymbol{H}, \beta) = K^{-N}\prod_{n=1}^{N}\prod_{k=1}^{K} N(\boldsymbol{x}_n|\boldsymbol{y}_k, \beta^{-1}\boldsymbol{I})^{z_{Kn}}, \qquad \boldsymbol{y}_k = \boldsymbol{f}(\boldsymbol{u}_k) = g(\boldsymbol{u}_k)\boldsymbol{h}(\boldsymbol{u}_k),$$

where $\boldsymbol{g}$ is a vector $g(\boldsymbol{u}_k)$ $(k = 1, \cdots, K)$ and $\boldsymbol{H}$ is a matrix $\boldsymbol{h}(\boldsymbol{u}_k)$ $(k = 1, \cdots, K)$. The prior distribution of $g(\boldsymbol{u})$ is given as a truncated GP with mean 0 and covariance function $c_g(\boldsymbol{u}_i, \boldsymbol{u}_j; \boldsymbol{\xi}_g)$, which handles positive-bounded random functions. The prior distribution of the dth entry $h_d(\boldsymbol{u})$ of $\boldsymbol{h}(\boldsymbol{u})$ is given as a GP with mean 0 and covariance function $c_h(\boldsymbol{u}_i, \boldsymbol{u}_j)$. The prior distributions of the parameters g and $\boldsymbol{H}$ are given by

$$p(\boldsymbol{g}) = N^{+}\left(\boldsymbol{g}\middle|\boldsymbol{0}, \boldsymbol{C}_g(\boldsymbol{\xi}_g)\right), \tag{2.2}$$

$$p(\boldsymbol{H}|r) = \prod_{d=1}^{D} N\left(\boldsymbol{h}_{(d)}|\boldsymbol{0}, \boldsymbol{C}_h\right), \tag{2.3}$$

where N^+ is a truncated normal distribution that handles positive-bounded random functions, $\boldsymbol{h}_{(d)}$ is a vector of the dth entry of the matrix $\boldsymbol{H}'$, and $\boldsymbol{C}_h$ is a matrix that consists of covariance function $c_h(\boldsymbol{u}_i, \boldsymbol{u}_j)$ as an element. Specifically, the covariance functions $c_g(\boldsymbol{u}_i, \boldsymbol{u}_j; \boldsymbol{\xi}_g)$ and $c_h(\boldsymbol{u}_i, \boldsymbol{u}_j)$ are given by

$$c_g\left(\boldsymbol{u}_i, \boldsymbol{u}_j; \boldsymbol{\xi}_g\right) = v_g \cdot \exp\left(-\frac{\left\|\boldsymbol{u}_i - \boldsymbol{u}_j\right\|^2}{2l_g}\right), \tag{2.4}$$

$$c_h\left(\boldsymbol{u}_i, \boldsymbol{u}_j\right) = \left\{\frac{2l(\boldsymbol{u}_i)l(\boldsymbol{u}_j)}{l^2(\boldsymbol{u}_i) + l^2(\boldsymbol{u}_j)}\right\}^{\frac{L}{2}} \exp\left(-\frac{\left\|\boldsymbol{u}_i - \boldsymbol{u}_j\right\|^2}{l^2(\boldsymbol{u}_i) + l^2(\boldsymbol{u}_j)}\right). \tag{2.5}$$

In equation (2.4), the hyperparameter $\boldsymbol{\xi}_g$ consists of v_g and l_g, referred to as the variance and the length-scale respectively, which control the magnitude of variances and smoothness of a positive-valued function $g(\boldsymbol{u})$ generated from the GP. In equation (2.5), the length-scale parameter $l(\boldsymbol{u})$ is a function of $\boldsymbol{u}$ and parameterized as $l(\boldsymbol{u}) = \exp\left(r(\boldsymbol{u})\right)$ with the function $r(\boldsymbol{u})$ following GP with mean 0 and covariance function $c_r(\boldsymbol{u}_i, \boldsymbol{u}_j; \boldsymbol{\xi}_r)$. Finally, the prior distribution of the precision parameter β is given by

$$p(\beta) = \mathrm{Gam}\left(\beta\middle|d_{\beta 0}, s_{\beta 0}\right), \tag{2.6}$$

where $\mathrm{Gam}(\cdot\,|d, s)$ denotes the gamma distribution, and its density function is defined by

$$\mathrm{Gam}(x|d, s) = \frac{s^d}{\Gamma(d)} x^{d-1}\exp(-sx),$$

where Γ is the gamma function $\Gamma(x) = \int_0^\infty e^{-t}t^{x-1}\,dt$.

The probability model of GTM-LDLV is defined as mentioned above. Here, the prior distribution of the non-liner function $f_d(\boldsymbol{u})$ (the dth entry of $\boldsymbol{f}(\boldsymbol{u})$) is derived as a GP with mean 0 and covariance function $c_f(\boldsymbol{u}_i, \boldsymbol{u}_j) = g(\boldsymbol{u}_i)g\left(\boldsymbol{u}_j\right)c_h\left(\boldsymbol{u}_i, \boldsymbol{u}_j\right)$. This covariance function shows that the GTM-LDLV model can control the variance and smoothness of the nonlinear function locally according to the value of the latent variable.

The unknown parameter to be estimated is $\boldsymbol{\theta} = \{\boldsymbol{Z}, \beta, \boldsymbol{g}, \boldsymbol{H}, \boldsymbol{r}\}$. In GTM-LDLV, the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{X})$ is approximately evaluated using a Markov Chain Monte Carlo (MCMC) method [101]. Iteratively sampling from the full conditional posterior distribution for each member of $\{\boldsymbol{Z}, \beta, \boldsymbol{g}, \boldsymbol{H}, \boldsymbol{r}\}$, we obtain a set of ensembles that follow the posterior distribution approximately. By taking the ensemble average over the samples from $p(\boldsymbol{\theta}|\boldsymbol{X})$, the parameters of GTM-LDLV are estimated. The simultaneous distribution of data $\boldsymbol{X}$ and parameters $\boldsymbol{\theta}$ is given by

$$p(\boldsymbol{X}, \boldsymbol{\theta}) = p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{g}, \boldsymbol{H}, \beta)p(\beta)p(\boldsymbol{g})p(\boldsymbol{H}|\boldsymbol{r})p(\boldsymbol{r}). \tag{2.7}$$

From equation (2.7) and Bayesian theorem, the posterior distribution of the latent variable $\boldsymbol{Z}$ is given by

$$p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}_{-\boldsymbol{Z}}) \propto p(\boldsymbol{X}, \boldsymbol{\theta}) \propto p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{g}, \boldsymbol{H}, \beta) \propto \prod_{n=1}^{N}\prod_{k=1}^{K} \exp\left(-\frac{\beta}{2}\|\boldsymbol{x}_n - \boldsymbol{y}_k\|^2\right)^{z_{kn}}, \tag{2.8}$$

where $\boldsymbol{\theta}_{-\boldsymbol{A}}$ represents a set of the parameters obtained by removing $\boldsymbol{A}$ from $\boldsymbol{\theta}$. Since summation over k of $\boldsymbol{Z}$ for each n is equal to 1, equation (2.8) can be written as

$$p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}_{-\boldsymbol{Z}}) = \prod_{n=1}^{N}\prod_{k=1}^{K} \gamma_k(\boldsymbol{x}_n; \boldsymbol{g}, \boldsymbol{H}, \beta)^{z_{kn}}, \tag{2.9}$$

where $\gamma_k(\boldsymbol{x}_n)$ is the probability that $\boldsymbol{x}_n$ is generated by the kth mixing element given $\boldsymbol{X}$ and $\boldsymbol{\theta}_{-\boldsymbol{Z}}$. $\gamma_k(\boldsymbol{x}_n)$ is given by

$$\gamma_k(\boldsymbol{x}_n; \boldsymbol{g}, \boldsymbol{H}, \beta) = \frac{\exp\left(-\frac{\beta}{2}\|\boldsymbol{x}_n - \boldsymbol{y}_k\|^2\right)}{\sum_{k'=1}^{K}\exp\left(-\frac{\beta}{2}\|\boldsymbol{x}_n - \boldsymbol{y}_{k'}\|^2\right)}. \tag{2.10}$$

Next, from equation (2.10) and Bayesian theorem, the conditional posterior distribution for parameters $\beta, \boldsymbol{g}, \boldsymbol{H}$, is given by

$$p\left(\beta|\boldsymbol{X}, \boldsymbol{\theta}_{-\beta}\right) = \mathrm{Gam}\left(\beta\middle|d_\beta, s_\beta\right), \tag{2.11}$$

$$p(\boldsymbol{g}|\boldsymbol{X},\boldsymbol{\theta}_{-\boldsymbol{g}}) = N^{+}(\boldsymbol{g}|\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g), \tag{2.12}$$

$$p(\boldsymbol{H}|\boldsymbol{X},\boldsymbol{\theta}_{-\boldsymbol{H}}) = \prod_{d=1}^{D} N(\boldsymbol{h}_{(d)}|\boldsymbol{\mu}_{h,d},\boldsymbol{\Sigma}_h). \tag{2.13}$$

The parameters of the conditional posterior distribution for parameters $\beta, \boldsymbol{g}, \boldsymbol{H}$ are given by

$$d_\beta = d_{\beta 0} + \frac{ND}{2},$$

$$s_\beta = s_{\beta 0} + \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K} z_{kn}\|\boldsymbol{x}_n - \boldsymbol{y}_k\|^2,$$

$$\boldsymbol{\mu}_g = \beta\boldsymbol{\Sigma}_g \mathrm{diag}(\boldsymbol{Z}\boldsymbol{X}'\boldsymbol{H}),$$

$$\boldsymbol{\Sigma}_g = (\beta\boldsymbol{G}\boldsymbol{\Lambda}_g + \boldsymbol{C}_g(\boldsymbol{\xi}_g)^{-1})^{-1},$$

$$\boldsymbol{\mu}_{h,d} = \beta\boldsymbol{\Sigma}_h\boldsymbol{\Lambda}_g\boldsymbol{Z}\boldsymbol{x}_{(d)}, \qquad \boldsymbol{\Sigma}_h = (\beta\boldsymbol{G}\boldsymbol{\Lambda}_g^2 + \boldsymbol{C}_h^{-1})^{-1}.$$

Here, $\mathrm{diag}(\boldsymbol{A})$ represents a column vector that contains the diagonal elements of matrix $\boldsymbol{A}$, $\boldsymbol{G}$ is a diagonal matrix that contains $\sum_{n=1}^{N} z_{kn}$ as diagonal element, $\boldsymbol{\Lambda}_g$ is a diagonal matrix that contains $g(\boldsymbol{u}_k)^2$ as diagonal element, and $\boldsymbol{x}_{(d)}$ is a vector of the dth entry of matrix $\boldsymbol{X}'$.

The posterior distribution of $\boldsymbol{r}$ is given by

$$p(\boldsymbol{r}|\boldsymbol{X},\boldsymbol{\theta}_{-\boldsymbol{r}}) \propto p(\boldsymbol{X},\boldsymbol{\theta}) \propto p(\boldsymbol{H}|\boldsymbol{r})p(\boldsymbol{r}) \propto \exp\big(s(\boldsymbol{r})\big),$$

$$s(\boldsymbol{r}) = -\frac{D}{2}\ln|\boldsymbol{C}_h| - \frac{1}{2}\sum_{d=1}^{D}\boldsymbol{h}'_{(d)}\boldsymbol{C}_h^{-1}\boldsymbol{h}_{(d)} - \frac{1}{2}\boldsymbol{r}'\boldsymbol{C}_r(\boldsymbol{\xi}_r)^{-1}\boldsymbol{r}. \tag{2.14}$$

Since $\boldsymbol{C}_h$ is a matrix that depends on $\boldsymbol{r}$, a sampling of $\boldsymbol{r}$ is performed as follows using Metropolis-Hasting method [101]. First, find the local maximum point $\hat{\boldsymbol{r}}$ of the log-likelihood function $s(\boldsymbol{r})$; then, generate the candidate point $\boldsymbol{r}^*$ from the proposed distribution $N(\boldsymbol{r}|\boldsymbol{m}_r,\boldsymbol{V}_r)$. Here, $\boldsymbol{m}_r,\boldsymbol{V}_r$ are given by

$$\boldsymbol{m}_r = \hat{\boldsymbol{r}} + \boldsymbol{V}_r\frac{\partial s(\boldsymbol{r})}{\partial \boldsymbol{r}}\bigg|_{\boldsymbol{r}=\hat{\boldsymbol{r}}}, \qquad \boldsymbol{V}_r = \left\{-\frac{\partial^2 s(\boldsymbol{r})}{\partial \boldsymbol{r}\partial \boldsymbol{r}'}\right\}^{-1}_{\boldsymbol{r}=\hat{\boldsymbol{r}}}.$$

When the current point is $\boldsymbol{r}^{t-1}$, the candidate point $\boldsymbol{r}^*$ is accepted with the next probability.

$$\min\left\{\frac{\exp\big(s(\boldsymbol{r}^*)\big)\,N(\boldsymbol{r}^{t-1}|\boldsymbol{m}_l,\boldsymbol{V}_l)}{\exp\big(s(\boldsymbol{r}^{t-1})\big)\,N(\boldsymbol{r}^*|\boldsymbol{m}_l,\boldsymbol{V}_l)},1\right\}. \tag{2.15}$$

A summary of the learning algorithm of GTM-LDLV for parameter estimation is shown in Algorithm 2.1. In the next section, we introduce periodic table generator (PTG) as an extension of GTM-LDLV and show the result of application of PTG to the element data.

Algorithm 2.1 GTM-LDLV

1: Prepare initial value $\boldsymbol{\theta}^0 = \{\boldsymbol{Z}^0, \beta^0, \boldsymbol{g}^0, \boldsymbol{H}^0, \boldsymbol{r}^0\}$.
for $t = 1$ to T **do**
Sample $\boldsymbol{Z}^t$ from $p(\boldsymbol{Z}|\boldsymbol{X}, \beta^{t-1}, \boldsymbol{g}^{t-1}, \boldsymbol{H}^{t-1}, \boldsymbol{r}^{t-1})$.
Sample β^t from $p(\beta|\boldsymbol{X}, \boldsymbol{Z}^t, \boldsymbol{g}^{t-1}, \boldsymbol{H}^{t-1}, \boldsymbol{r}^{t-1})$.
Sample $\boldsymbol{g}^t$ from $p(\boldsymbol{g}|\boldsymbol{X}, \boldsymbol{Z}^t, \beta^t, \boldsymbol{H}^{t-1}, \boldsymbol{r}^{t-1})$.
Sample $\boldsymbol{H}^t$ from $p(\boldsymbol{H}|\boldsymbol{X}, \boldsymbol{Z}^t, \beta^t, \boldsymbol{g}^t, \boldsymbol{r}^{t-1})$.
Sample $\boldsymbol{r}^t$ from $p(\boldsymbol{r}|\boldsymbol{X}, \boldsymbol{Z}^t, \beta^t, \boldsymbol{g}^t, \boldsymbol{H}^t)$.
end for
For a sufficiently large number T_b, record $\boldsymbol{\theta}^t = \{\boldsymbol{Z}^t, \beta^t, \boldsymbol{g}^t, \boldsymbol{H}^t, \boldsymbol{r}^t\}, t = T_b, T_b + 1, \cdots, T$.

The model parameters of GTM-LDLV $\boldsymbol{\theta}^{ldlv} = \{\boldsymbol{Z}^{ldlv}, \beta^{ldlv}, \boldsymbol{g}^{ldlv}, \boldsymbol{H}^{ldlv}, \boldsymbol{r}^{ldlv}\}$ are estimated by taking the average of $\boldsymbol{\theta}^t = \{\boldsymbol{Z}^t, \beta^t, \boldsymbol{g}^t, \boldsymbol{H}^t, \boldsymbol{r}^t\}$ for $t = T_b, T_b + 1, \cdots, T$.

3 Recreation of the periodic table using an unsupervised machine learning algorithm

As briefly explained earlier, herein, we seek to answer whether machine learning can reproduce or recreate the periodic table by using observed physicochemical properties of the elements. Thus, we developed a periodic table generator (PTG), which is an unsupervised machine learning algorithm based on GTM that can automate the translation of high-dimensional data into a tabular form with varying layouts on-demand. PTG can autonomously produce various arrangements of chemical symbols, which organize a two-dimensional array such as Mendeleev's periodic table or three-dimensional spiral table according to the underlying periodicity in the given data. We further show what the PTG learned from the element data and how the element features, such as melting point and electronegativity, are compressed to the lower-dimensional latent spaces.

3.1 Introduction

In this study, we created various periodic tables using a machine learning algorithm. The dataset that we used consisted of 39 features (melting points, electronegativity, and so on) of 54 elements with the atomic numbers 1-54, corresponding to hydrogen to xenon (Fig. 3.1 for the heatmap display). A wide variety of dimensionality reduction methods have been proposed so far, such as principal component analysis (PCA), kernel PCA [76], isometric feature mapping (ISOMAP) [77], local linear embedding (LLE) [78], and t-distributed stochastic neighbor embedding (t-SNE) [79]. However, none of these methods can adequately visualize the underlying periodic laws (Fig. 3.2). To begin with, none of these methods offer a tabular representation. The task of building a periodic table can be regarded as dimension reduction of the element data to arbitrary given "discrete" points rather than a continuous space. Kernelized sorting [102] has been proposed as a method that can provide a tabular representation of data. It achieves data visualization by maximizing the dependency between matched pairs of high-dimensional data points and low-dimensional lattice points by means of the Hilbert Schmidt Independence Criterion [102]. A visualization result of the element data on the 2-dimensional 6×9 rectangular lattice using kernelized sorting is shown in Fig. 3.3. As shown in Fig. 3.3, elements in each period of the standard periodic table are nearly configured in a fan shape from the bottom left to the top right, but the table fails to capture the discontinuity from group 18 to group 1 as in the original table. Therefore, we developed a new unsupervised machine learning algorithm called the periodic table generator (PTG), which relies on GTM [7] with latent variable dependent length-scale and variance (GTM-LDLV) [11]. One of the advantages of using GTM-LDLV arises from its ability to represent complex response surfaces. Elemental data shows a complex response surface on the feature space. Controlling the two hyperparameters, the GTM-LDLV can flexibly represent functions whose smoothness and amplitude vary locally in the feature space. With this model, we automate the process of translating patterns of high-dimensional feature vectors to an arbitrary given layout of lower-dimensional point clouds.

PTG produces various arrangements of chemical symbols, which organize, for example, a two-dimensional array such as Mendeleev's table or three-dimensional spiral table according to the underlying periodicity in the given data. We will show what the machine intelligence learned from the given data and how the element features were compressed to the reduced dimensionality representations. The periodic tables can also be regarded as the most primitive descriptor of chemical elements. Hence, we will highlight the representation capability of such element-level descriptors in the description of materials that are used in machine learning tasks of materials property prediction.

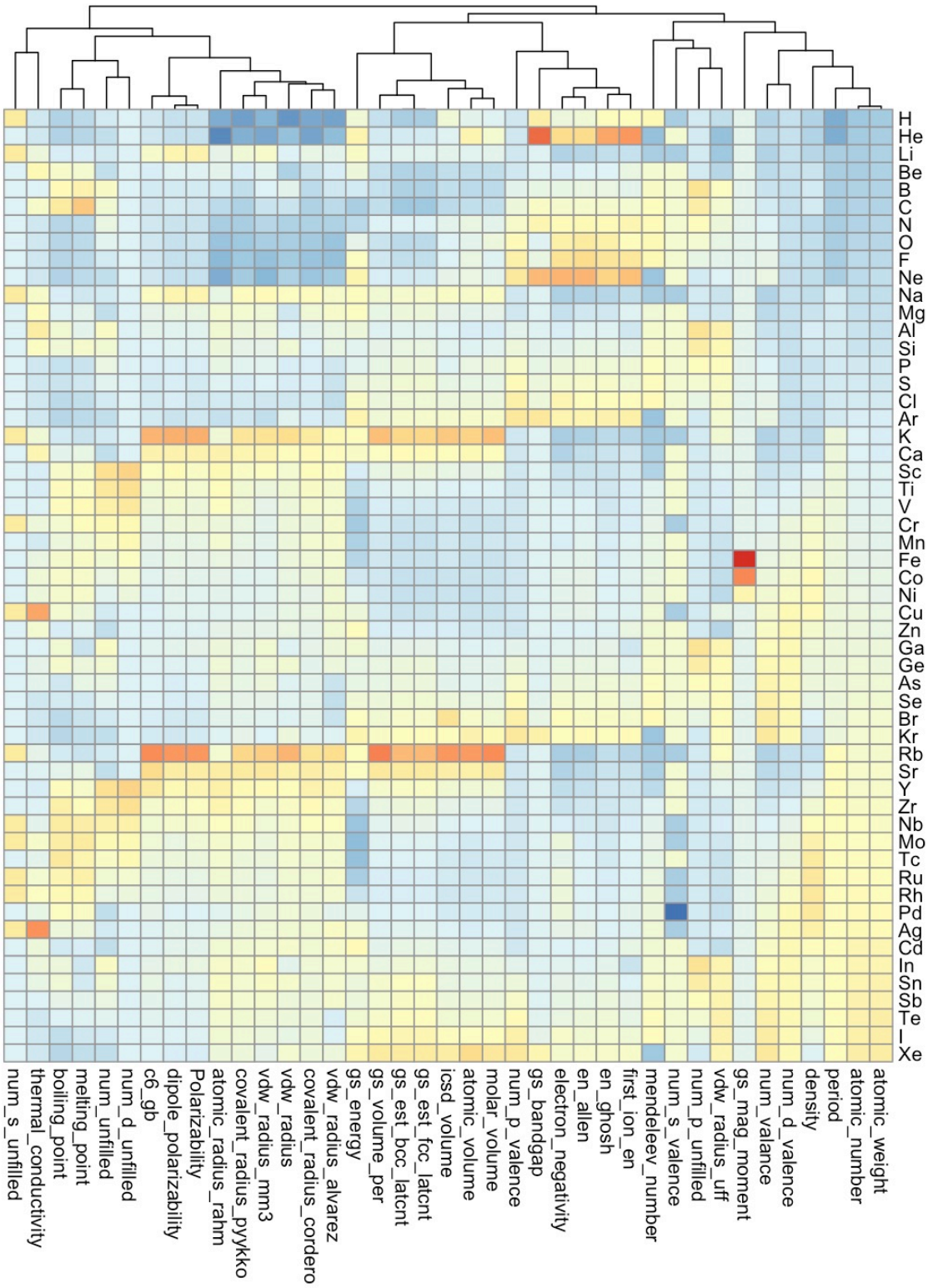


Figure 3.1. Heatmap of the element data used in this study. The data matrix is clustered for each column (features).

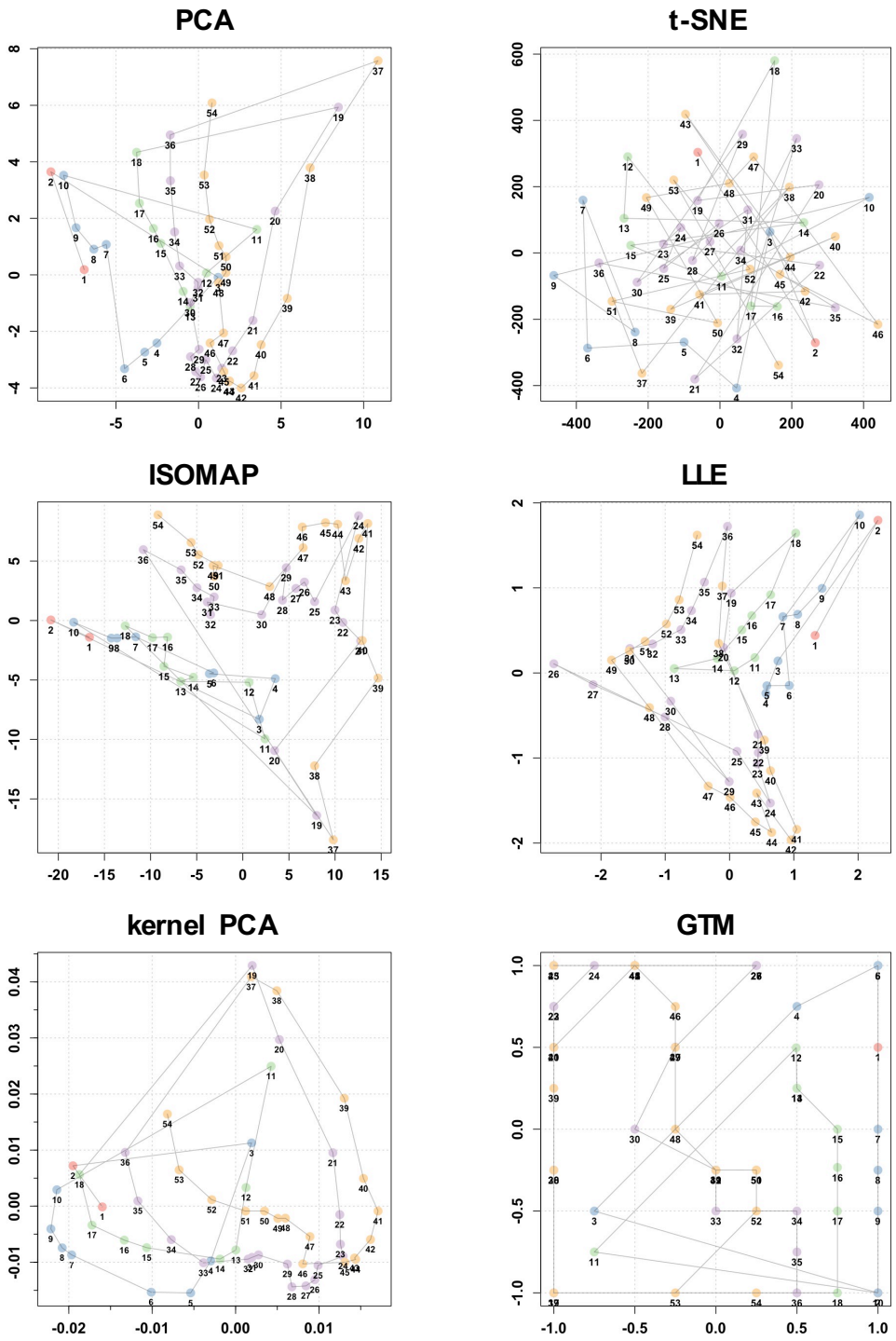


Figure 3.2. Visualization results of the element data on the two-dimensional space using PCA (top-left), t-SNE (top-right), ISOMAP with neighbors = 3 (middle-left), LLE with neighbors = 9 (middle-right), kernel PCA with ANOVA kernel and sigma = 0.2 (bottom-left), and GTM with $K = 9 \times 9$ grid points and 16 basis functions (bottom-right). The elements are color-coded by periods and numbered by atomic numbers. A line passing through the elements is drawn in the order of atomic numbers.

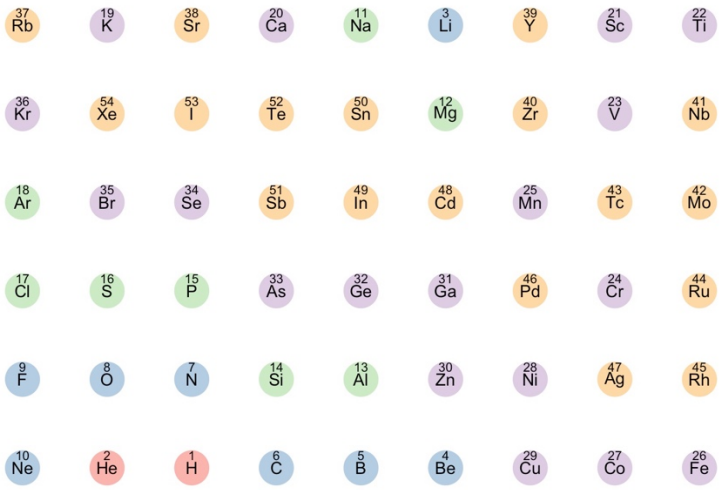


Figure 3.3. Layout of the 54 elements on the 6×9 rectangular lattice using kernelized sorting. The elements are color-coded by periods and numbered by atomic numbers.

3.2 Methods

3.2.1 Computational workflow

The workflow of the PTG begins by specifying a set of point clouds, called "nodes" hereafter, in a low-dimensional latent space to which chemical elements with observed physicochemical features are assigned. The nodes can take any positional structure such as equally spaced grid points on a rectangular for an ordinal table, spiral, cuboid, cylinder, cone, and so on. A Gaussian process (GP) model [14] is used to map the predefined nodes to the higher-dimensional feature space in which the element data are distributed. A trained GP defines a manifold in the feature space to be fitted with respect to the observed element data. The smoothness of the manifold is governed by a specified covariance function called the kernel function, which associates the similarity of nodes in the latent space with that in the feature space. The estimated GP defines a posterior probability or responsibility of each chemical element belonging to one of the nodes. An element is assigned to one node with the highest posterior probability.

As indicated by the failure of some existing methods of statistical dimension reduction, such as PCA, t-SNE, and LLE, the manifold surface of the mapping from chemical elements to their physiochemical properties is highly complex. Therefore, we adopted GTM-LDLV as a model of PTG, which is a GTM that can model locally varying smoothness in the manifold. To ensure non-overlapping assignments such that no multiple elements share the same node, we operated the GTM-LDLV with the constraint of one-to-one matching between nodes and elements. To satisfy this, the number of nodes K has to be larger than the number of elements N. However, direct learning with $K > N$ suffers from high computational costs and unstable estimation performance. Specifically, the use of redundant nodes leads to many suboptimal solutions corresponding to undesirable matchings to the chemical elements. To alleviate this problem, PTG was designed to take a three-step procedure (Fig. 3.4) that relies on a coarse-to-fine strategy. In the first step, we operated the training of GTM-LDLV with a small set of nodes such that $K < N$. In the following step, we generated additional nodes such that $K > N$, and the expanded node-set was transferred to the feature space by performing the interpolative prediction made by the given GTM-LDLV. Finally, the pretrained model was fine-tuned subject to the one-to-one matching between the N elements and the K nodes for tabular construction. The procedure for each step is detailed below.

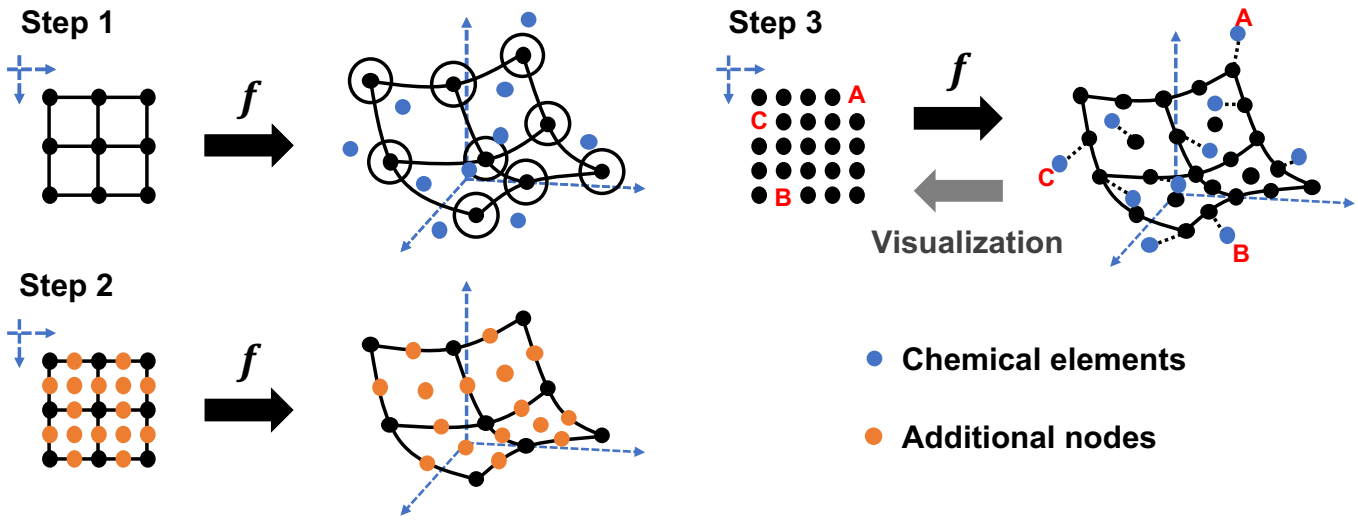


Figure 3.4. Workflow of PTG that relies on a three-step coarse-to-fine strategy to reduce the occurrence of undesirable matching between chemical elements and redundant nodes.

Step 1 (GTM-LDLV): The first step of the PTG is the same as the original GTM-LDLV. In GTM-LDLV, K nodes, $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_K$, arbitrarily arranged in the L-dimensional latent space are first prepared. Then, we build a nonlinear function $\boldsymbol{f}(\boldsymbol{u}_k)$ that maps the predefined nodes to the D-dimensional feature space. The model $\boldsymbol{f}(\boldsymbol{u}_k)$ defines an L-dimensional manifold in the D-dimensional feature space, which is fitted with respect to the N data points of element features. The dimension of the latent space is set to $L \leq 3$ for visualization.

It is assumed that the D-dimensional feature vector $\boldsymbol{x}_n$ of element n is generated independently from a mixture of K Gaussian distributions, where the mixing rates are all equal to $1/K$, and the mean and the covariance matrix of each distribution are $\boldsymbol{y}_k = \boldsymbol{f}(\boldsymbol{u}_k)$ and $\beta^{-1}\boldsymbol{I}$, respectively ($\boldsymbol{I}$ denotes the identity matrix). According to GTM-LDLV, the mean $\boldsymbol{f}(\boldsymbol{u}_k)$ is modeled to be the product of two functions, a D-dimensional vector-valued function $\boldsymbol{h}(\boldsymbol{u}_k)$ and a positive scalar function $g(\boldsymbol{u}_k)$. Here, we introduce a vector of K latent variables $\boldsymbol{z}_n = (z_{1n}, \cdots, z_{Kn})'$, which indicates the assignment of element n to one of the given K nodes. The kth entry z_{kn} takes the value of 1 if $\boldsymbol{x}_n$ is generated by the kth component distribution, and 0 otherwise. Here, let $\boldsymbol{X}$ denote a matrix of $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N$ of the elements and $\boldsymbol{Z}$ be a matrix of $\boldsymbol{z}_1, \cdots, \boldsymbol{z}_N$. Then, their joint distribution is given by

$$p(\boldsymbol{X}, \boldsymbol{Z} | \boldsymbol{g}, \boldsymbol{H}, \beta) = K^{-N} \prod_{n=1}^{N} \prod_{k=1}^{K} N(\boldsymbol{x}_n | \boldsymbol{y}_k, \beta^{-1}\boldsymbol{I})^{z_{kn}}, \tag{3.1}$$

$$\boldsymbol{y}_k = \boldsymbol{f}(\boldsymbol{u}_k) = g(\boldsymbol{u}_k)\boldsymbol{h}(\boldsymbol{u}_k), \tag{3.2}$$

where $N(\cdot | \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the Gaussian density function with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, $\boldsymbol{g}$ is a vector of $g(\boldsymbol{u}_k)$ $(k = 1, \cdots, K)$, and $\boldsymbol{H}$ is a matrix of $\boldsymbol{h}(\boldsymbol{u}_k)$ $(k = 1, \cdots, K)$.

The prior distribution of $g(\boldsymbol{u})$ is given as a truncated GP with mean 0 and covariance function $c_g(\boldsymbol{u}_i, \boldsymbol{u}_j; \boldsymbol{\xi}_g)$, which handles positive-bounded random functions. The prior distribution of the dth entry $h_d(\boldsymbol{u})$ of $\boldsymbol{h}(\boldsymbol{u})$ is given as a GP with mean 0 and covariance function $c_h(\boldsymbol{u}_i, \boldsymbol{u}_j)$. To be specific, the covariance functions, $c_g(\boldsymbol{u}_i, \boldsymbol{u}_j; \boldsymbol{\xi}_g)$ and $c_h(\boldsymbol{u}_i, \boldsymbol{u}_j)$, are given by

$$c_g(\boldsymbol{u}_i, \boldsymbol{u}_j; \boldsymbol{\xi}_g) = v_g \cdot \exp\left(-\frac{\|\boldsymbol{u}_i - \boldsymbol{u}_j\|^2}{2l_g}\right), \tag{3.3}$$

$$c_h(\boldsymbol{u}_i, \boldsymbol{u}_j) = \left\{\frac{2l(\boldsymbol{u}_i)l(\boldsymbol{u}_j)}{l^2(\boldsymbol{u}_i) + l^2(\boldsymbol{u}_j)}\right\}^{\frac{L}{2}} \exp\left(-\frac{\|\boldsymbol{u}_i - \boldsymbol{u}_j\|^2}{l^2(\boldsymbol{u}_i) + l^2(\boldsymbol{u}_j)}\right). \tag{3.4}$$

In equation (3.3), the hyperparameter $\boldsymbol{\xi}_g$ consists of v_g and l_g, referred to as the variance and the length-scale, which respectively control the magnitude of variances and smoothness of a positive-valued function $g(\boldsymbol{u})$ generated from the GP. In equation (3.4), the length-scale parameter $l(\boldsymbol{u})$ is a function of $\boldsymbol{u}$ and parameterized as $l(\boldsymbol{u}) = \exp(r(\boldsymbol{u}))$ with the function $r(\boldsymbol{u})$ following the GP with mean 0 and covariance function $c_r(\boldsymbol{u}_i, \boldsymbol{u}_j; \boldsymbol{\xi}_r)$. Finally, a gamma prior is placed on the precision

parameter β in equation (3.1).

The covariance function in equation (3.4) is the key in GTM-LDLV. In general, a covariance function in a GP governs a degree of preservation between the similarity of any inputs, for example, $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$, and the similarity of their outputs. The heterogeneous variance over the latent space in equation (3.4) can bring locally varying smoothness in resulting manifolds in the feature space. In addition, the variance function is statistically estimated with the hierarchically specified GP prior based on the covariance function $c_r(\boldsymbol{u}_i, \boldsymbol{u}_j; \boldsymbol{\xi}_r)$.

The unknown parameter to be estimated is $\boldsymbol{\theta} = \{\boldsymbol{Z}, \beta, \boldsymbol{g}, \boldsymbol{H}, \boldsymbol{r}\}$. In GTM-LDLV, the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{X})$ is approximately evaluated using a Markov Chain Monte Carlo (MCMC) method. Iteratively sampling from the full conditional posterior distribution for each $\{\boldsymbol{Z}, \beta, \boldsymbol{g}, \boldsymbol{H}, \boldsymbol{r}\}$, we obtained a set of ensembles that follow the posterior distribution approximately. By taking the ensemble average over the samples from $p(\boldsymbol{\theta}|\boldsymbol{X})$, the parameters of the GTM-LDLV are estimated. A detailed description of the GTM-LDLV is already described in Section 2.10.

Step 2 (node expansion): To avoid the occurrence of improper assignments of the N elements to a redundant set of nodes, we adopt a coarse-to-fine strategy. Starting from an initially trained GP model of $K < N$ at step 1, we refine the model with an increased number of nodes $K \geq N$. For example, 5×5 nodes evenly arranged on the area $[-1, 1] \times [-1, 1]$ at step 1 are incremented to $K = 9 \times 9$ by placing additional nodes at middle points of the line segments connecting between each node. With the currently given parameters, we can infer the values of $r(\boldsymbol{u})$ of the covariance function in equation (3.4) at the expanded nodes, $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_K$. Likewise, the values of $g(\boldsymbol{u})$ and $\boldsymbol{h}(\boldsymbol{u})$ are interpolated. By performing such initialization, we proceed to the next round of GTM-LDLV.

Step 3 (GTM-LDLV subject to one-to-one assignments): Finally, the resulting GTM-LDLV is fine-tuned to obtain a tabular display by running the above procedure subject to a one-to-one matching between the N elements and the K nodes. By definition, the conditional posterior distribution of the assignment variables is represented as

$$p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}_{-\boldsymbol{Z}}) \propto \prod_{n=1}^{N} \prod_{k=1}^{K} \exp\left(-\frac{\beta}{2}\|\boldsymbol{x}_n - \boldsymbol{y}_k\|^2\right)^{z_{kn}} = \exp\left(-\frac{\beta}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} z_{kn}\|\boldsymbol{x}_n - \boldsymbol{y}_k\|^2\right),$$

where $\boldsymbol{\theta}_{-\boldsymbol{A}}$ represents a set of the parameters obtained by removing $\boldsymbol{A}$ from $\boldsymbol{\theta}$. In the MCMC calculation in step 1, we iteratively draw a sample of $\boldsymbol{Z}$ from this distribution. Here, instead of performing the random sampling, we conduct the maximization of the logarithmic posterior with respect to $\boldsymbol{Z}$ subject to the constraint of one-to-one assignments. The problem amounts to finding the solution of

$$\max_{\boldsymbol{Z} \in A} -\sum_{n=1}^{N} \sum_{k=1}^{K} z_{kn}\|\boldsymbol{x}_n - \boldsymbol{y}_k\|^2,$$

$$A = \left\{\boldsymbol{Z} \,\middle|\, \sum_{k=1}^{K} z_{kn} = 1 \ (n = 1, \cdots, N), \quad \sum_{n=1}^{N} z_{kn} \leq 1 \ (k = 1, \cdots, K)\right\}.$$

This is regarded as a transportation problem where the sum of the squared Euclidean distance between an element feature $\boldsymbol{x}_n$ and a node $\boldsymbol{y}_k$ embedded in the feature space is the cost of transporting one item from source k to destination n under constraint A. We used the lpSolve package [103] in R [104] to solve the transportation problem.

This partially modified MCMC was iterated few times (e.g., $T = 10$) to make a fine-tuning of the currently given parameters. The assignment variables and the other parameters that exhibited the highest likelihood were chosen to form the final estimate of the PTG. A summary of the PTG algorithm is shown in Algorithm 3.1.

Algorithm 3.1 Periodic Table Generator (PTG)

1: Prepare initial value $\boldsymbol{\theta}^0 = \{\boldsymbol{Z}^0, \beta^0, \boldsymbol{g}^0, \boldsymbol{H}^0, \boldsymbol{r}^0\}$.
for $t = 1$ to T **do**
Sample $\boldsymbol{Z}^t$ from $p(\boldsymbol{Z}|\boldsymbol{X}, \beta^{t-1}, \boldsymbol{g}^{t-1}, \boldsymbol{H}^{t-1}, \boldsymbol{r}^{t-1})$.
Sample β^t from $p(\beta|\boldsymbol{X}, \boldsymbol{Z}^t, \boldsymbol{g}^{t-1}, \boldsymbol{H}^{t-1}, \boldsymbol{r}^{t-1})$.
Sample $\boldsymbol{g}^t$ from $p(\boldsymbol{g}|\boldsymbol{X}, \boldsymbol{Z}^t, \beta^t, \boldsymbol{H}^{t-1}, \boldsymbol{r}^{t-1})$.
Sample $\boldsymbol{H}^t$ from $p(\boldsymbol{H}|\boldsymbol{X}, \boldsymbol{Z}^t, \beta^t, \boldsymbol{g}^t, \boldsymbol{r}^{t-1})$.
Sample $\boldsymbol{r}^t$ from $p(\boldsymbol{r}|\boldsymbol{X}, \boldsymbol{Z}^t, \beta^t, \boldsymbol{g}^t, \boldsymbol{H}^t)$.
end for
For a sufficiently large number T_b, record $\boldsymbol{\theta}^t = \{\boldsymbol{Z}^t, \beta^t, \boldsymbol{g}^t, \boldsymbol{H}^t, \boldsymbol{r}^t\}, t = T_b, T_b + 1, \cdots, T$.

2: The model parameters of GTM-LDLV $\boldsymbol{\theta}^{ldlv} = \{\boldsymbol{Z}^{ldlv}, \beta^{ldlv}, \boldsymbol{g}^{ldlv}, \boldsymbol{H}^{ldlv}, \boldsymbol{r}^{ldlv}\}$ are estimated by taking the average of $\boldsymbol{\theta}^t = \{\boldsymbol{Z}^t, \beta^t, \boldsymbol{g}^t, \boldsymbol{H}^t, \boldsymbol{r}^t\}$ for $t = T_b, T_b + 1, \cdots, T$. Increase the number of nodes on the latent space so that $K \geq N$ is satisfied. Considering the parameters estimated by GTM-LDLV (the first step of PTG) as observation values, interpolate the parameters corresponding to the newly generated nodes using GP regression.

3: The parameters $\boldsymbol{\theta}^{itp} = \{\boldsymbol{Z}^{itp}, \beta^{itp}, \boldsymbol{g}^{itp}, \boldsymbol{H}^{itp}, \boldsymbol{r}^{itp}\}$ obtained as above are used as initial values for the next procedure.
for $t = 1$ to T' **do**
$\boldsymbol{Z}^t \leftarrow \underset{\boldsymbol{Z} \in A}{\operatorname{argmax}}\, p(\boldsymbol{Z}|\boldsymbol{X}, \beta^{t-1}, \boldsymbol{g}^{t-1}, \boldsymbol{H}^{t-1}, \boldsymbol{r}^{t-1}),\ A = \{\boldsymbol{Z} | \sum_{n=1}^{N} z_{kn} \leq 1\ (k = 1, \cdots, K)\}$.
$\beta^t \leftarrow \underset{\beta}{\operatorname{argmax}}\, p(\beta|\boldsymbol{X}, \boldsymbol{Z}^t, \boldsymbol{g}^{t-1}, \boldsymbol{H}^{t-1}, \boldsymbol{r}^{t-1})$.
$\boldsymbol{g}^t \leftarrow \underset{\boldsymbol{g}}{\operatorname{argmax}}\, p(\boldsymbol{g}|\boldsymbol{X}, \boldsymbol{Z}^t, \beta^t, \boldsymbol{H}^{t-1}, \boldsymbol{r}^{t-1})$.
$\boldsymbol{H}^t \leftarrow \underset{\boldsymbol{H}}{\operatorname{argmax}}\, p(\boldsymbol{H}|\boldsymbol{X}, \boldsymbol{Z}^t, \beta^t, \boldsymbol{g}^t, \boldsymbol{r}^{t-1})$.
$\boldsymbol{r}^t \leftarrow \underset{\boldsymbol{r}}{\operatorname{argmax}}\, p(\boldsymbol{r}|\boldsymbol{X}, \boldsymbol{Z}^t, \beta^t, \boldsymbol{g}^t, \boldsymbol{H}^t)$.
end for

3.2.2 Interpretation

PTG autonomously creates a tabular display of the chemical elements according to the estimated $\boldsymbol{Z}$. To understand how element features such as melting point and electronegativity are compressed on the low-dimensional tabular display, each of feature is mapped onto the resulting table. Specifically, we overlay a smoothed heatmap of each feature on the table. With this PTG property landscape [83], we can visually understand the distribution of the topographical mapping that indicates how the element features are embedded in the latent space.

3.2.3 Periodic table as an element descriptor

We considered an evaluation basis for the quality of a designed periodic table in terms of a novel view from data science. A periodic table, including Mendeleev's classic table, can be considered as one of the most primitive descriptors that encodes known element features into the coordinate system of a low-dimensional latent space. Neighboring elements on a table should behave similarly and possess similar physicochemical properties. Inspired by such an idea, we considered the use of a periodic table as a descriptor of chemical elements in a task of predicting materials properties based on machine learning [100]. The periodic table was then evaluated quantitatively based on the predictive performance of the descriptor.

For a given table, its coordinates $\boldsymbol{u}_{k(1)}, \cdots, \boldsymbol{u}_{k(N)}$ of the nodes to which the N elements are assigned were used as a set of element descriptors. For a compound S, its fraction of the N elements was denoted by $w_1(S), \cdots, w_N(S)$ where $0 \leq w_n(S) \leq 1$ and $\sum_{n=1}^{N} w_n(S) = 1$. The compositional descriptor of S was calculated by $\boldsymbol{\phi}(S) = \sum_{n=1}^{N} w_n(S)\, \boldsymbol{u}_{k(n)}$. With this descriptor, we derived a prediction model $Y = f(\boldsymbol{\phi}(S))$, which is trained in m training instances $\{Y_i, S_i\}_{i=1}^{m}$ and describes a physicochemical property Y as a function of the descriptor $\boldsymbol{\phi}(S)$ for any given compound S. Descriptors exhibiting higher predictability should be recognized as providing more efficient compression performances on the N elements. For comparison, the same analysis was performed using two-dimensional coordinates of the standard periodic table, PCA and t-SNE, respectively.

3.2.4 Data: element features

The element feature set was extracted from XenonPy [32, 33], which is a Python library for materials informatics, by using an application programming interface (API) (see the XenonPy website [33]). The original dataset consisted of 74 features of 118 elements. Since elements with large atomic numbers contained many missing values, we selected 54 elements with the atomic numbers 1–54 corresponding to hydrogen to xenon, which we considered sufficient to retain the periodic rule. After removing features that contained one or more missing values, the dataset was reduced to 39 features of 54 elements. For the 54×39 data matrix, each feature (column) was standardized to have mean 0 and variance 1. A heatmap of the data matrix and detailed description of the 39 features are provided in Fig 3.1 and Table 3.1, respectively.

Table 3.1. Detailed description of the 39 element-level features used in this analysis.

Element-level properties used for analysis

Feature	Description
atomic_number	Number of protons found in the nucleus of an atom
atomic_radius_rahm	Atomic radius by Rahm et al
atomic_volume	Atomic volume
atomic_weight	The mass of an atom
boiling_point	Boiling temperature
c6_gb	C_6 dispersion coefficient in a.u
covalent_radius_cordero	Covalent radius by Cerdero et al
covalent_radius_pyykko	Single bond covalent radius by Pyykko et al
density	Density at 295K
dipole_polarizability	Dipole polarizability
electron_negativity	Tendency of an atom to attract a shared pair of electrons
en_allen	Allen's scale of electronegativity
en_ghosh	Ghosh's scale of electronegativity
first_ion_en	First ionisation energy
gs_bandgap	DFT bandgap energy of T=0K ground state
gs_energy	DFT energy per atom (raw VASP value) of T=0K ground state
gs_est_bcc_latcnt	Estimated BCC lattice parameter based on the DFT volume
gs_est_fcc_latcnt	Estimated FCC lattice parameter based on the DFT volume
gs_mag_moment	DFT magnetic momenet of T=0K ground state
gs_volume_per	DFT volume per atom of T=0K ground state
icsd_volume	Atom volume in ICSD database
mendeleev_number	Atom number in mendeleev's periodic table
melting_point	Melting point
molar_volume	Molar volume
num_unfilled	Total unfilled electron
num_valence	Total valence electron
num_d_unfilled	Unfilled electron in d shell
num_d_valence	Valence electron in d shell
num_p_unfilled	Unfilled electron in p shell
num_p_valence	Valence electron in p shell
num_s_unfilled	Unfilled electron in s shell
num_s_valence	Valence electron in s shell
period	Period in the periodic table
thermal_conductivity	Thermal conductivity at 25 C
vdw_radius	Van der Waals radius
vdw_radius_alvarez	Van der Waals radius according to Alvarez
vdw_radius_mm3	Van der Waals radius from the MM3 FF
vdw_radius_uff	Van der Waals radius from the UFF
Polarizability	Ability to form instantaneous dipoles

3.2.5 Analysis procedure

We performed the PTG on two different layouts of nodes, square, and three-dimensional conical layouts. In the square layout of $L = 2$, we set $K = 25$ in the first step of PTG in which the 5×5 nodes were evenly arranged on the area $[-1, 1] \times [-1, 1]$. In the second step, we increased the number of nodes to 9×9 by placing new nodes at the middle points of the line segments connecting between each node. In the conical layout of $L = 3$, we first used a set of nodes with $K = 25$, which were arranged uniformly on the surface of the cone placed in the area $[-1, 1] \times [-1, 1] \times [-1, 1]$. The cone was sliced into 4 sections in the same height along the vertical axis. Then, 1 (vertex), 4, 8, and 12 (bottom) nodes were uniformly placed on the outer part of the 4 cut surfaces. In the next step, the number of slices was increased by 7, and 1 (vertex), 4, 8, 12, 16, 20, and 24 (bottom) nodes were uniformly arranged in the same way. In both the cases, we set $\boldsymbol{\xi}_g = \boldsymbol{\xi}_r = (1/3, 3)$, the number of iterations in MCMC was set to $T = 10{,}000$ with the burn-in step $T_b = 5{,}000$, and the number of iterations in the third step of fine-tuning was set to $T = 10$. See Section 3.5 for further details on the hyperparameter settings and analysis procedure.

The PTG algorithm was implemented using R codes, which are available at [105] with the element dataset. Readers can run the PTG algorithm with the element data used herein. As a demonstration, the PTG was performed on three other layouts: a rectangular table with 5×18 equally spaced grids, which is same as the layout of the standard periodic table, and two cylinder and cubic three-dimensional layouts. The results are shown in Fig. 3.17.

3.3 Results

3.3.1 Results of PTG

Square table

Fig. 3.5 shows the PTG-created layout of the 54 elements on the 9×9 square lattice. Elements in each period of the standard periodic table are configured in a fan shape from the top left to the bottom right. The elements in the square table are clearly separated into metals and non-metals by the red dashed line shown in Fig. 3.5. The 3d and 4d transition elements are separated and both clustered in the lower right. In addition, the elements are clearly clustered by groups such as alkali metals, alkaline earth metals, halogens, and noble gases. This looks like a variant of the original periodic table: the original table is folded around the center on which transition elements are positioned, the two separated blocks of groups 1-2 and 13-18 in the first to third periods are brought nearer to each other while keeping away from the area of transition elements, and they are stored in a square table. Notably, the square table exhibits the discontinuity from group 18 to group 1, as in the original table. Though results are not shown, the same discontinuity appears frequently in most square tables created in the experiments under different conditions.

Conical table

Fig. 3.6 shows the PTG-created layout on three-dimensional conical nodes. The elements are arranged in a spiral structure starting from the top of the cone according to increasing atomic numbers. Viewed from the top, the elements are stratified concentrically by the periods of the standard periodic table. This view is slightly similar to the circular periodic table that was constructed in a different study [97]. One block corresponded to a set of elements divided according to the orbital type of the electrons of the highest energy levels. In the standard periodic table, helium (He: circled by the red line in Fig. 3.7) is located away from the other s-block elements (a set of elements colored red in Fig. 3.7), but in the conical table, it is located close to them. It was also seen that the elements in the conical table are clearly classified into typical elements and transition elements by the red line shown in Fig. 3.7. A blank space was observed between groups 1 and 18 on the conical table implying that there is a gap of properties between them in the feature space.

In the spiral structure viewed from above, the atomic numbers are monotonically arranged from top to bottom except for a few elements. The disorder appears in group 6 to 7: manganese (Cr: atomic number = 24) and iron (Mn: 25) in period 4 or molybdenum (Mo: 42) and technetium (Tc: 43) in period 5. In the conical table, the elements are arranged radially according to groups, and elements of groups 1 and 2 are located a little away from group 3.

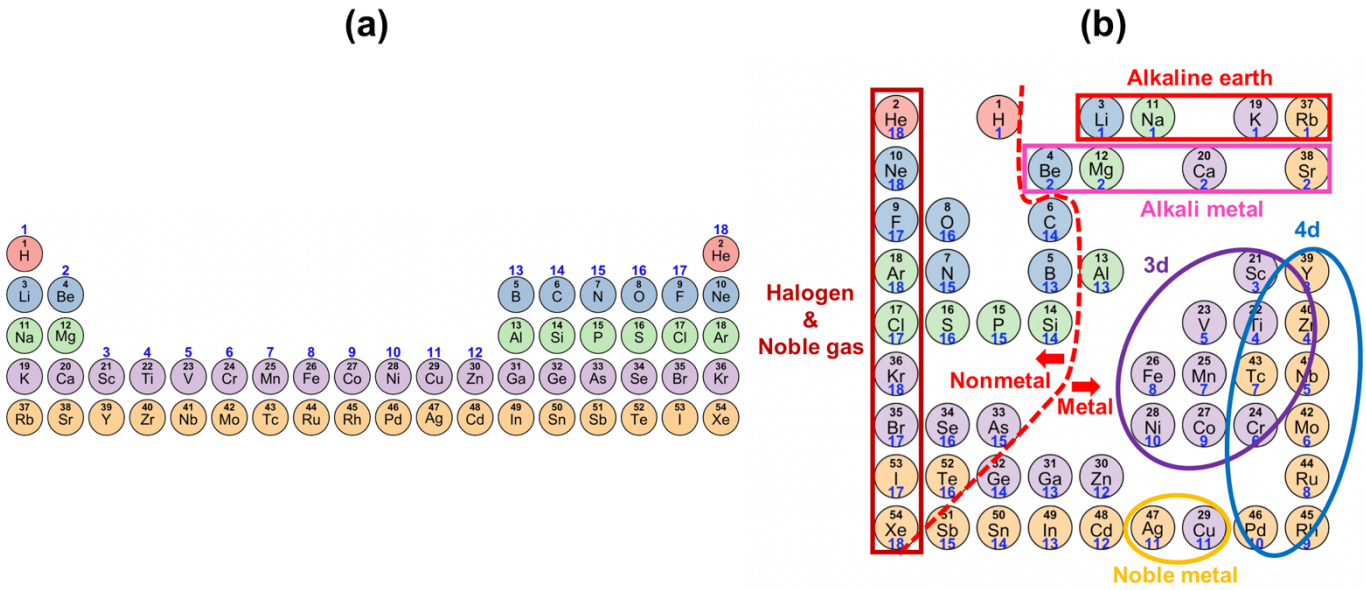


Figure 3.5. (a) The currently most common periodic table of the elements. (b) Square PTG table created from the training data of 39 features of the 54 elements. The elements are color-coded by periods and numbered as per their atomic number. The number shown in blue below each element symbol represents the group number (the column in the standard periodic table).

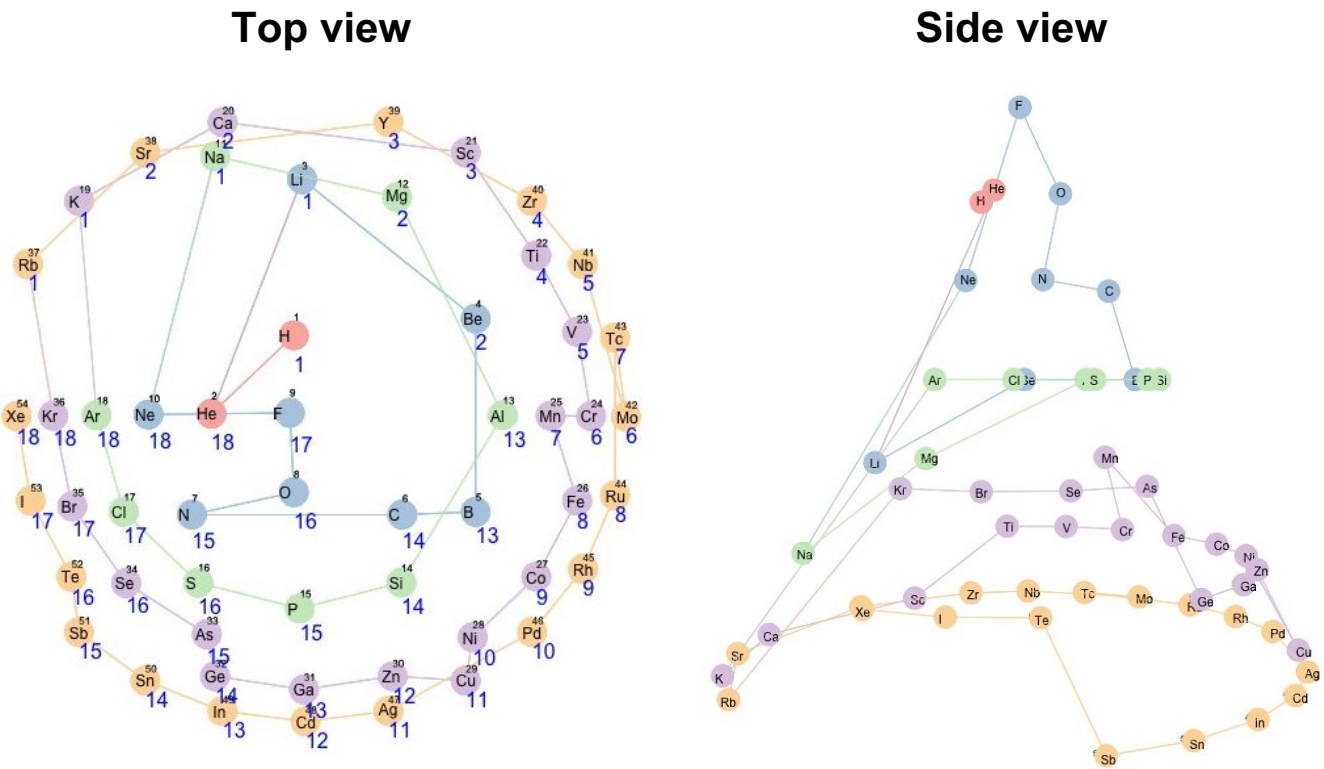


Figure 3.6. PTG-created conical table of 54 chemical elements. The elements are color-coded according to five periods and numbered as per their atomic number. A line passing through the elements is drawn in the order of atomic numbers. The number shown in blue below each element symbol represents the group number (the column in the standard periodic table). The left and right figures show the same table viewed from top and side, respectively.

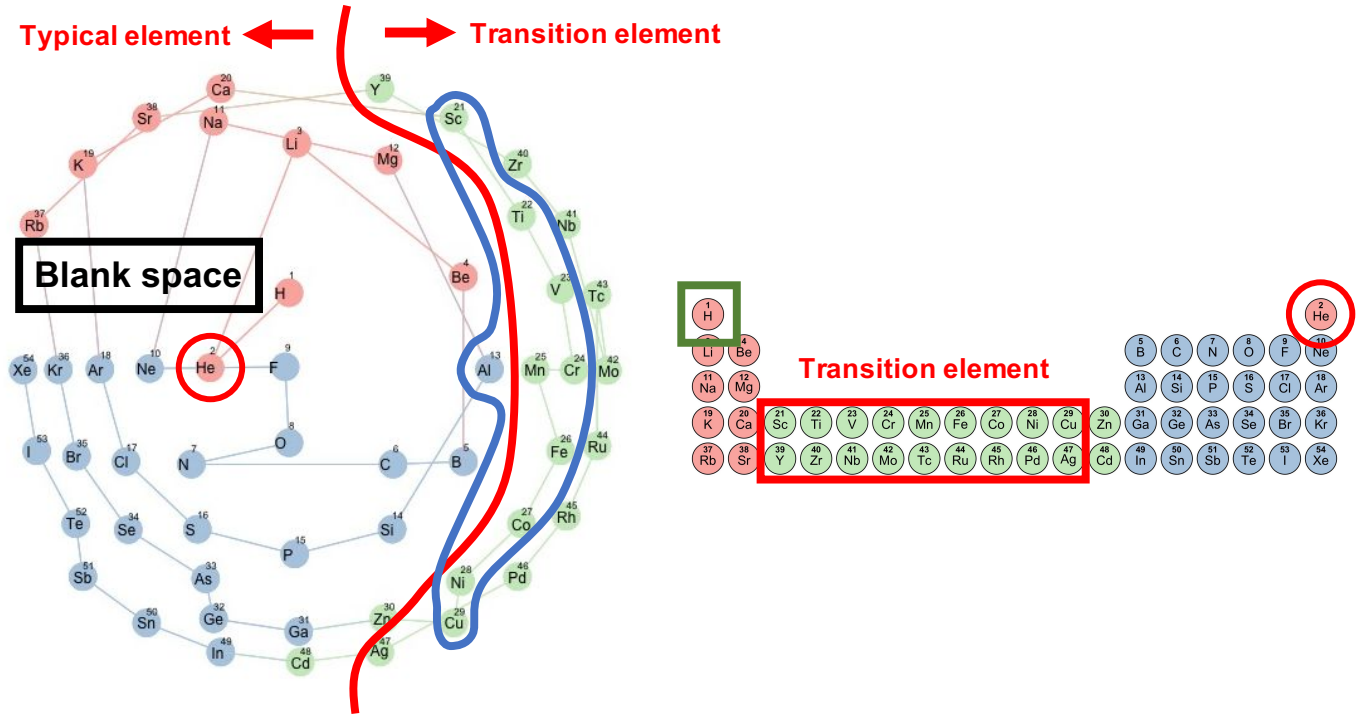


Figure 3.7. The left panel shows a conical table viewed from above. The elements are color-coded according to three blocks in the standard periodic table that are indicated in the right panel. The red line on the left indicates the segment between transition elements and typical elements.

3.3.2 Interpretation

To understand how the element features have been embedded on the created tables, each feature was mapped on the lower-dimensional latent space (Fig. 3.8). In the property landscape of the conical table, atomic radius increases gradually and concentrically from the top of the cone, electron negativity decreases gradually and concentrically from the top of the cone, and melting point gradually increases from right to left. The distribution of thermal conductivity is a little more complicated than the former three, but continuity and unimodality is still held on the surface of the three-dimensional conical table. On the contrary, in the square table, the landscapes of some element features, e.g., atomic radius and thermal conductivity, exhibit multimodality. This discontinuity arises from the unnatural layout of the elements in the two-dimensional tabular representation, as in the standard periodic table. The PTG property landscapes of the 39 features are shown in Figs. 3.9 and 3.10, respectively.

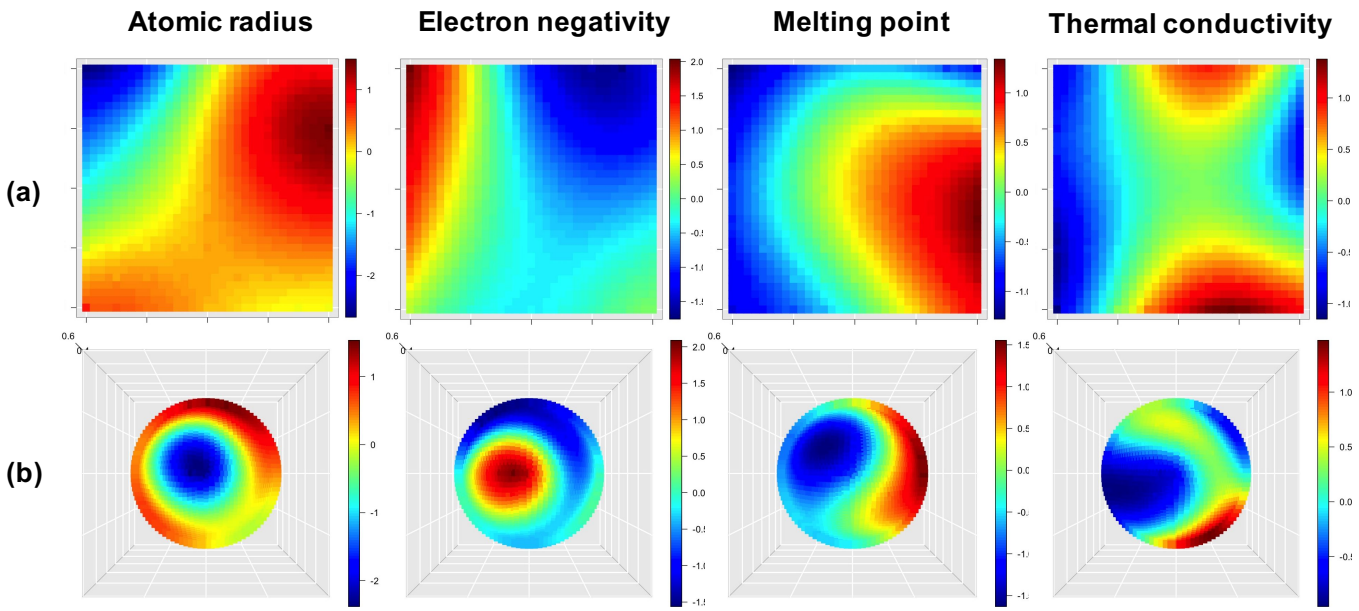
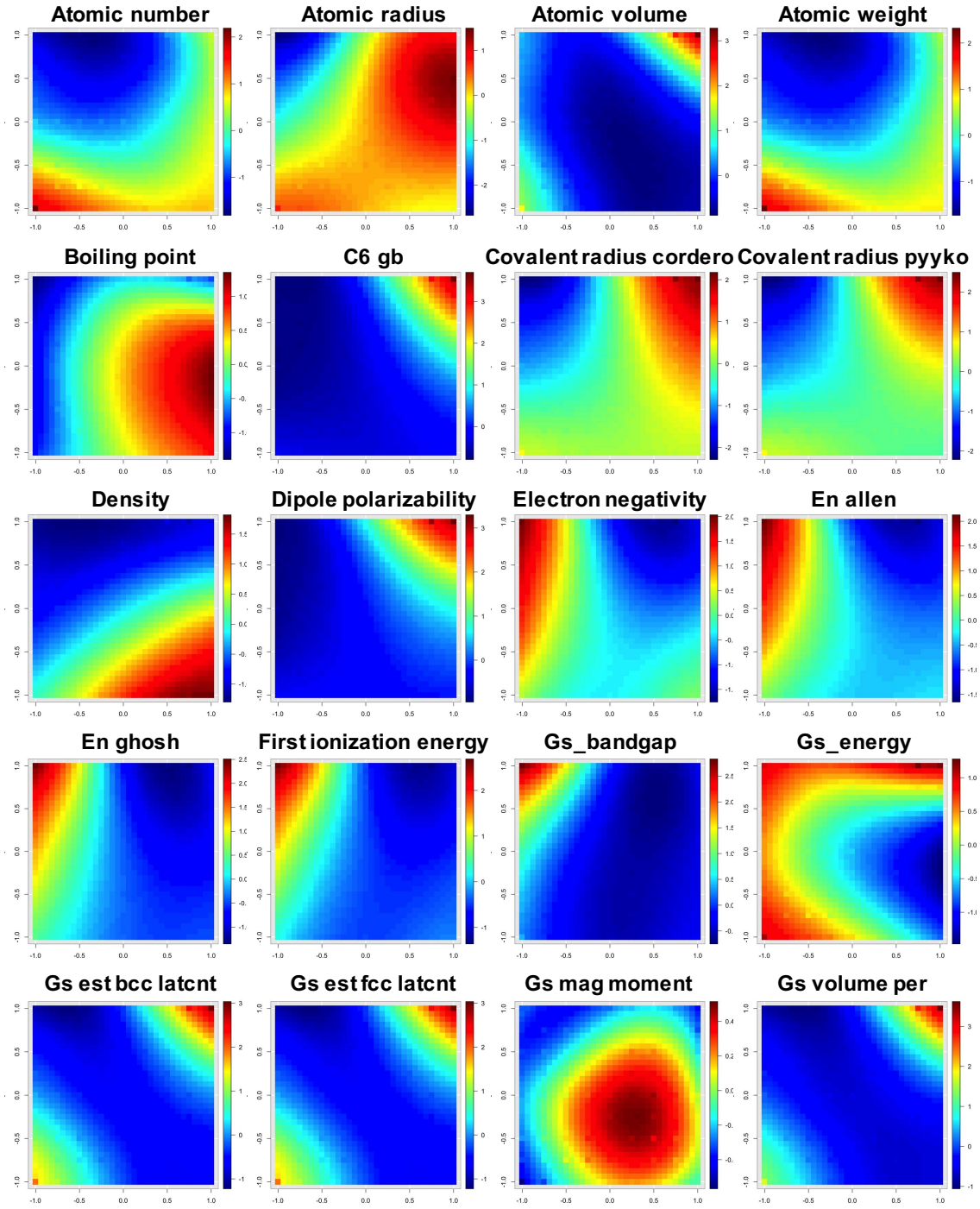


Figure 3.8. Property landscapes of atomic radius (Rahm et al. [106]), electron negativity, melting point, and thermal conductivity at 25°C that are embedded in the latent spaces. The heatmaps are laid on (a) the square table in Fig. 3.5 and (b) the conical table (top view) in Fig. 3.6.



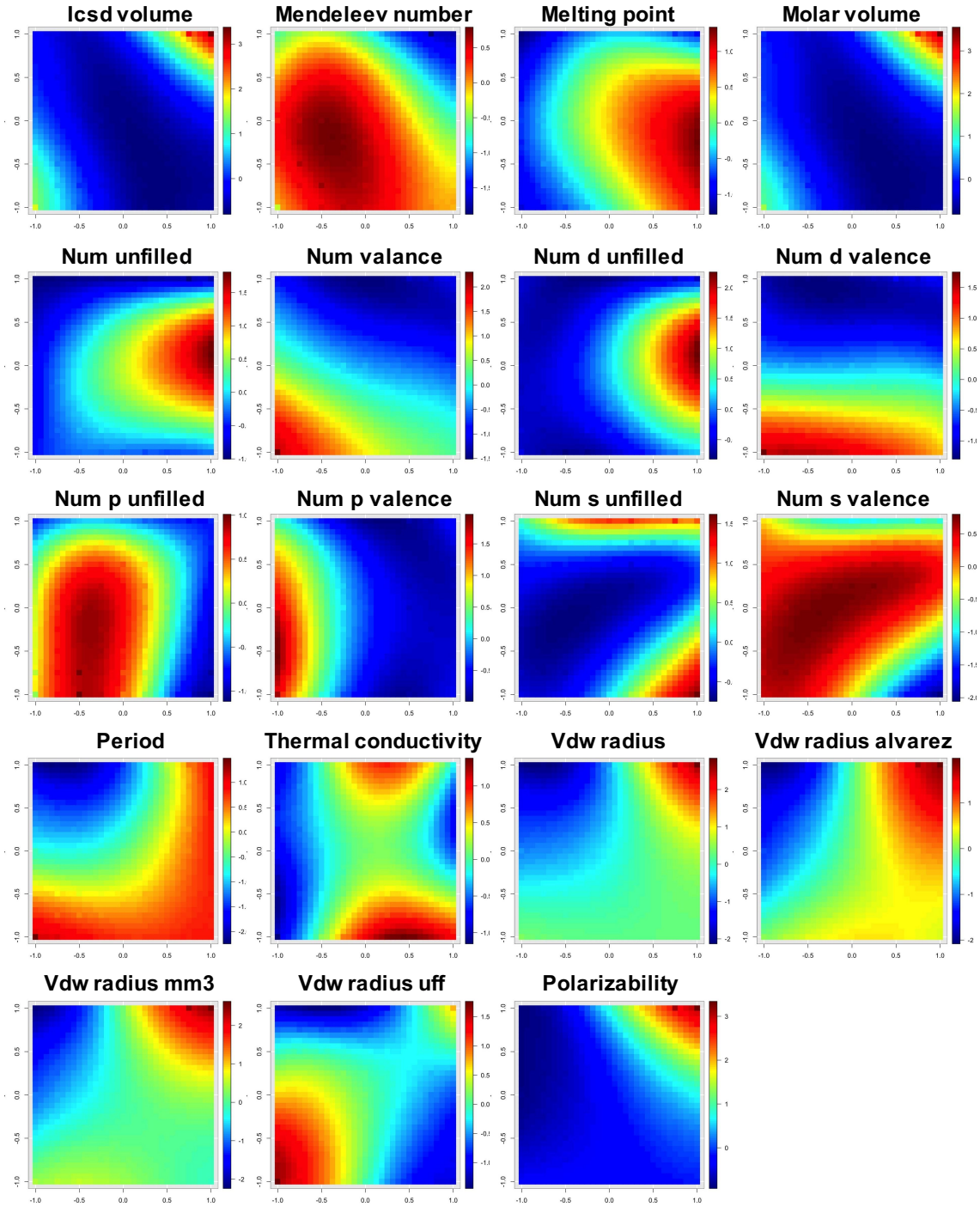
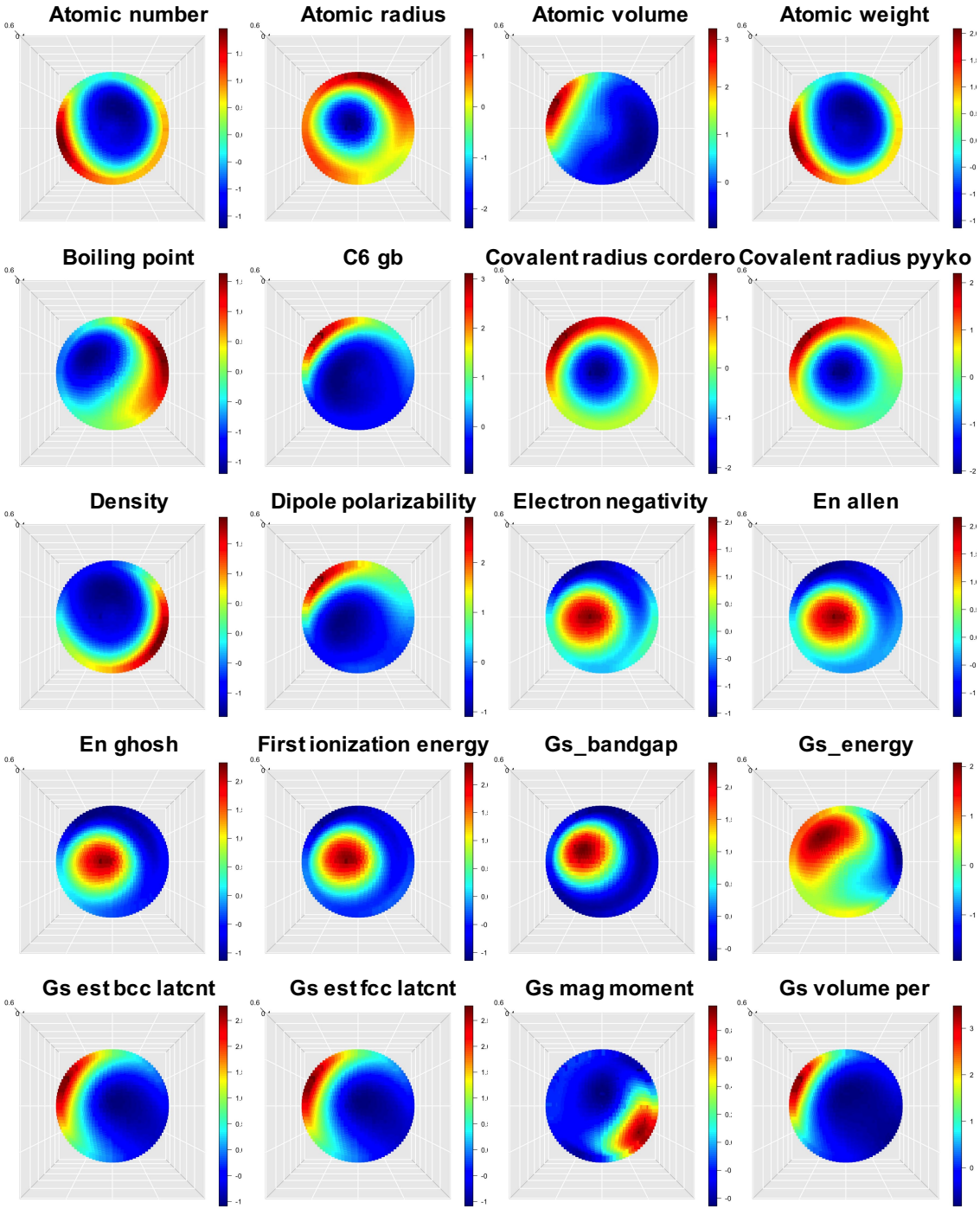


Figure 3.9. PTG property landscape of all 39 features for the square PTG table shown in Fig. 3.5.


Atomic number
Atomic radius
Atomic volume
Atomic weight
Boiling point
C6 gb
Covalent radius cordero
Covalent radius pyyko
Density
Dipole polarizability
Electron negativity
En allen
En ghosh
First ionization energy
Gs_bandgap
Gs_energy
Gs est bcc latcnt
Gs est fcc latcnt
Gs mag moment
Gs volume per

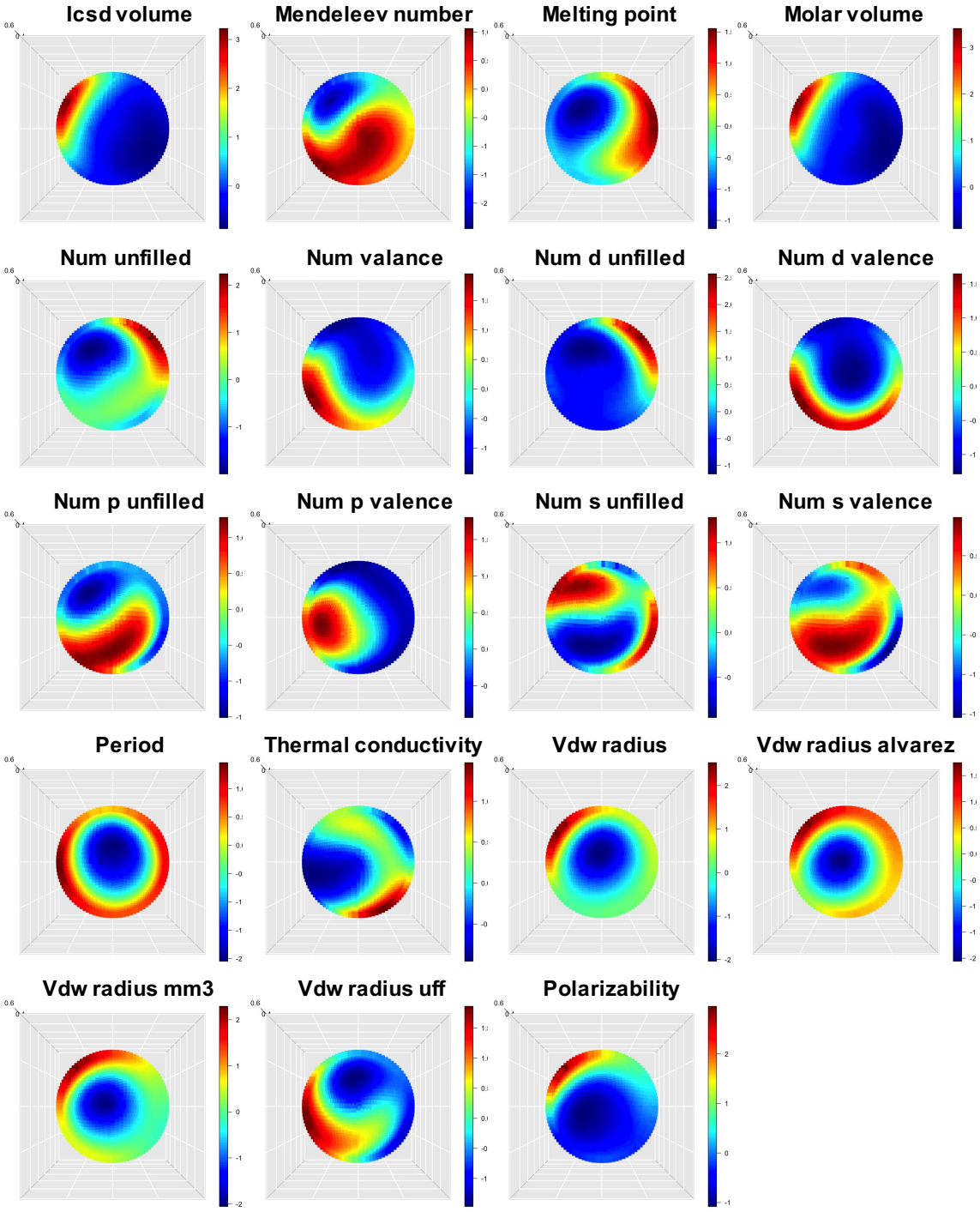


Figure 3.10. PTG property landscapes of all 39 features for the conical PTG table shown in Fig. 3.6.

3.3.3 Quantitative comparison of periodic tables

To evaluate the validity of a periodic table and uncover the information gain and loss in the reduced representation, we considered the use of a table as an element descriptor in machine learning tasks. The task to be addressed was the prediction of formation energies of inorganic compounds. The dataset that we used for the training of random forest regressors (RF) [107] was obtained from the Materials Project [15]. Among all inorganic compounds in the Materials Project, we selected compounds that are stable and consist of elements with the atomic number 1-54 (H to Xe). The dataset consisted of the formation energies per atom of 12,373 inorganic compounds.

The objective here was to train an RF that describes the formation energy as a function of the conical descriptor $\boldsymbol{\phi}(S)$ obtained by composing S and the three-dimensional coordinates of the elements in the conical table. This is described in Section 3.2. For comparison, we built four different models using the descriptors based on the two-dimensional coordinates in the created square table, the standard periodic table, PCA, and t-SNE, respectively.

We performed five-fold cross-validation on the 12,373 samples for the six types of descriptors. As shown in Fig. 3.11, the conical PTG achieved a mean absolute error (MAE) of 0.464 eV/atom and a root mean square error (RMSE) of 0.643 eV/atom, whereas the MAE and RMSE of the square PTG and the standard periodic table were 0.533 eV/atom and 0.719 eV/atom, and 0.549 eV/atom and 0.734 eV/atom, respectively. The models based on PCA and t-SNE showed MAE of 0.631 eV/atom and 0.667 eV/atom, respectively, and RMSE of 0.830 eV/atom and 0.859 eV/atom, respectively, which clearly shows that they were less accurate in their predictions. Finally, the model based on the complete set of the 39-dimensional feature showed MAE of 0.197 eV/atom and RMSE of 0.311 eV/atom (this shows how the overall information is being retained by the tables). In summary, the square PTG is slightly superior to the standard periodic table, but the conical PTG table outperforms the standard periodic table, the square PTG, PCA, and t-SNE, respectively.

A detailed investigation of the prediction results provided some insights into the difference in information compression between the three-dimensional conical table and the standard periodic table. We focused on a subset of the compounds used in the validation, hereafter denoted by D_{cone} (i.e., the conical descriptor dominant set), that had MAE values less than 0.3 eV/atom for the conical descriptor but 1.0 eV/atom greater than the conical descriptor for the standard periodic table. Likewise, we identified D_{standard} with MAE values less than 0.3 eV/atom for the standard periodic table but 1.0 eV/atom greater than the standard periodic table for the conical table. We counted the frequency of a chemical element in D_{cone} and D_{standard}, and evaluated the enrichment of the element by comparing its expected frequency calculated with the background, i.e., the number of occurrences in the overall population (12,373 compounds in Materials Project). As shown in Fig. 3.12, a significantly enriched group in D_{cone} comprises transition elements in the fourth period that correspond to atomic number 21-29. Aluminum (Al) is also enriched in D_{cone} (Fig. 3.12: set of elements circled by a blue line). Notably, these over-represented elements form a cluster in the created conical table (Fig. 3.7: set of elements circled by a blue line). On the contrary, hydrogen (H) is significantly enriched in D_{standard} (Fig. 3.12: element circled by green line). H is located just above lithium (Li) in the standard periodic table (Fig. 3.7: element circled by a green line), while it is located between fluorine (F) and Li in the conical periodic table.

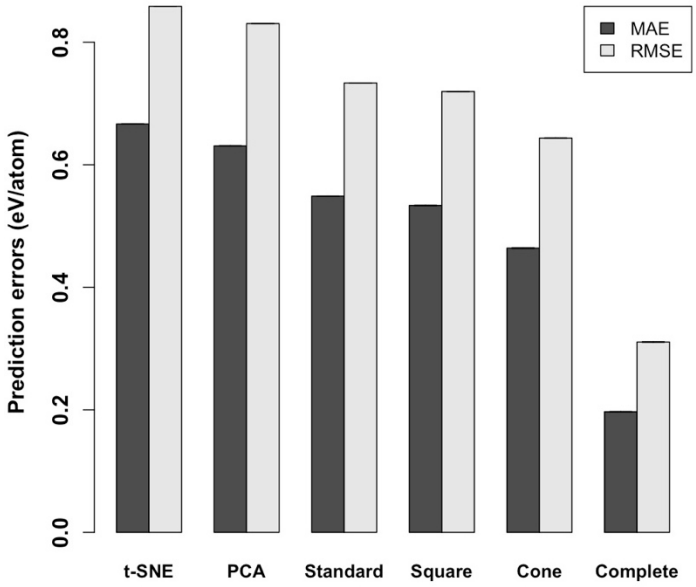


Figure 3.11. Prediction performance of the formation energy per atom for the models using six different descriptors. The vertical axis indicates cross-validated MAE and RMSE of RF regressors trained with the six different descriptors obtained from the coordinates of elements in the representation made by t-SNE and PCA (corresponding to top-left and top-right in Fig. 3.2, respectively), the standard periodic table, the square PTG table, the conical PTG table, and the complete set of the 39-dimensional feature that were used to build the PTG table, respectively. The error bars denote the standard deviations in five independent trials of the cross-validation (they are invisible because of substantially small scales).

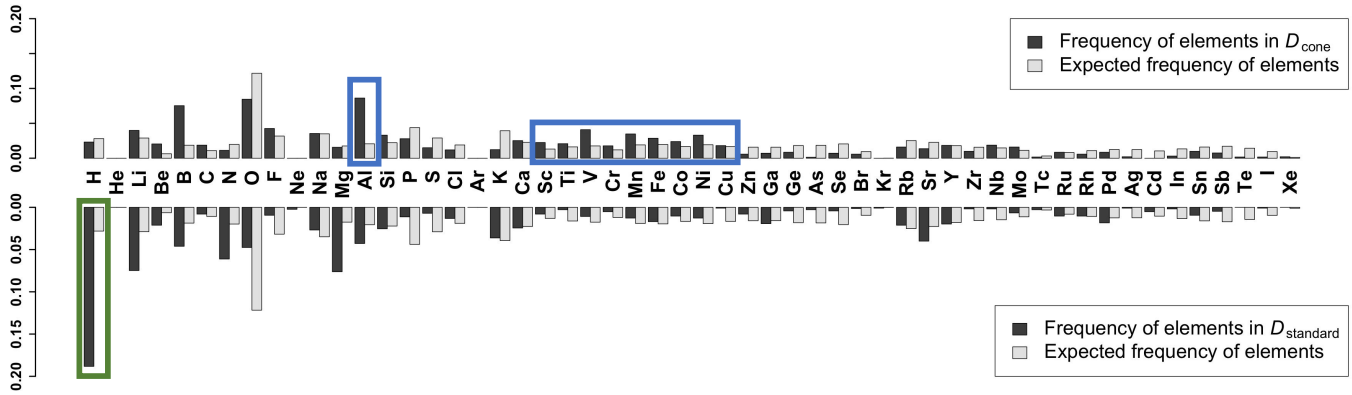


Figure 3.12. Comparison of the frequencies of chemical elements in D_{cone} (top: black bar chart) and D_{standard} (bottom: black bar chart). White bar charts show the expected frequency calculated with the number of occurrences in the overall population.

3.4 Estimation of the intrinsic dimension of element data

When visualizing data by dimension reduction methods, it is important to know the intrinsic dimension of the data. Thus, at first, the intrinsic dimension was estimated for the entire elemental data (assuming the entire data set has the same intrinsic dimension), using the maximum likelihood (MLE) method [108], the k-nearest neighbor (kNN) method [109], and DANCo (dimensionality from angle and norm concentration) method [110]. The result of the dimension estimators m_k by these three methods with varying number of neighbors $k \in \{3, \ldots, 25\}$ is shown in Fig. 3.13. From the results of Fig. 3.13, if the entire dataset has the same intrinsic dimension, it is estimated that the intrinsic dimension of the element data will be 3 or 4.

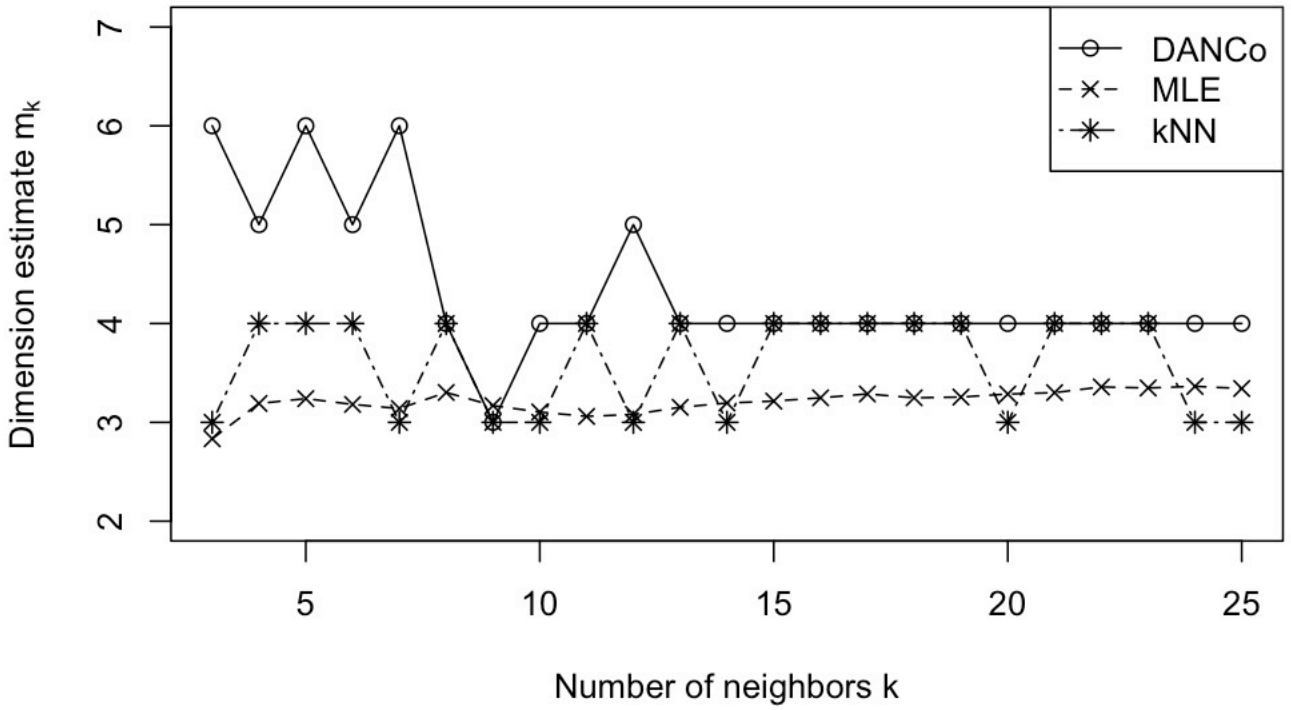


Figure 3.13. Dimension estimators m_k as a function of k by DANCo, MLE, and kNN methods.

Next, we estimated the local intrinsic dimension for each data point of the element data (i.e., each element) using OTPM (optimally topology preserving feature map) method [111]. In OTPM, estimation of the intrinsic dimension is based on local PCA of the pointers of the nodes in the OTPM and their direct neighbors. Since the dimension estimator for each data point m_k^n varies depending on the number of nodes k, the final dimension estimator for each data point $\hat{m}^n$ is estimated as follows:

$$\hat{m}^n = \frac{1}{6}\sum_{k=5}^{10} m_k^n.$$

The distributions of the dimension estimator through the standard periodic table, the square PTG table shown in Fig. 3.5, and the conical PTG table shown in Fig. 3.6 (top view and side view) are shown in Fig. 3.14. According to Fig. 3.14, the dimension estimator for each data point, that is, for each element, varies from approximately 1.0 to 3.5, suggesting that the entire element data do not have the same intrinsic dimension. Furthermore, according to Fig. 3.14, it can be seen that a group of elements with high dimension estimators (consisting of Li, Be, B, C, Mg, Al, Si, P, and S) are separated in the standard periodic table but clustered in the square PTG table. From Fig. 3.14, it can be seen that the tables produced by PTG are clustering the elements according to their local intrinsic dimensions.

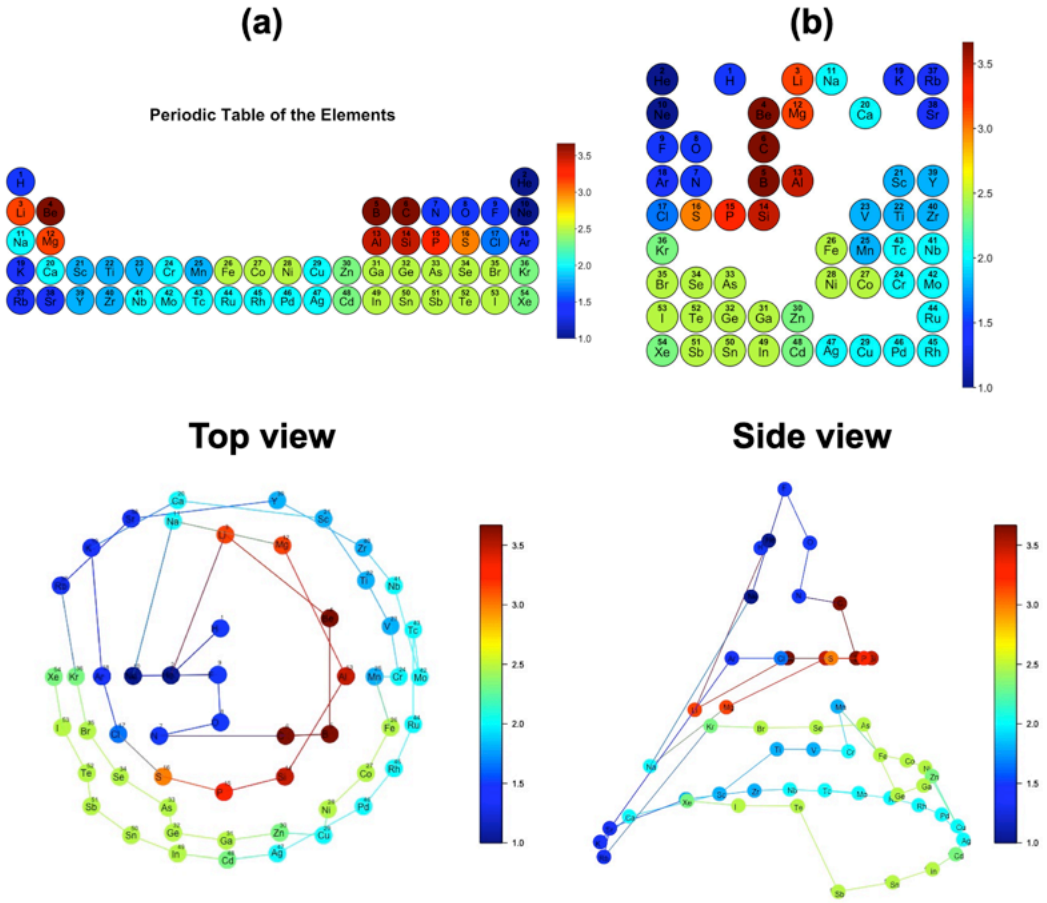


Figure 3.14. Distribution of the dimension estimator $\widehat{m}^n$ through (a) the standard periodic table, (b) the square PTG table shown in Fig. 3.5, and the conical PTG table shown in Fig. 3.6. The bottom left and bottom right figures show the same conical PTG table viewed from the top and side, respectively.

3.5 Notes on the PTG Algorithm

It should be noted that PTG may produce different visualization results for each trial even under the same hyperparameter settings. Indeed, PTG with element data produced different tables for each trial of the algorithm. This implies that PTG reached a different local maxima on the likelihood surface for each trial. PTG tries to fit lower-dimensional manifolds to the shape of data cloud, and there are multiple solutions to this. Therefore, it is expected that there are many local maxima that are separated from one other on the likelihood surface of PTG. This is not counterintuitive as there should not be a unique optimal solution for arranging elements in the new periodic table. One way to deal with this problem is to run the algorithm multiple times under the same hyperparameter settings and enumerate multiple visualization results. The final result should then be selected from the list of obtained tables based on some selection criterion.

In step 1 of PTG with the element data, it was observed that the learning of the model became unstable and was terminated when the non-information prior distribution was used as prior distribution of the precision β. To address the problem, a prior distribution of β with a small scale and a sufficiently large rate was used. This prior distribution keeps the variance β^{-1} estimated from the posterior distribution larger than a certain value, and it made the learning stable. In the next section, we introduce details of the analysis procedure and hyperparameter settings used in this study.

3.6 Details of analysis procedure

We performed PTG on two different node layouts namely, square and three-dimensional conical layouts. In the square layout of $L = 2$, we set $K = 25$ in the first step of PTG in which the 5×5 nodes were evenly arranged on the area $[-1, 1] \times [-1, 1]$. In the second step, we increased the number of nodes to 9×9 by placing new nodes at middle points on

the line segments connecting between each node. In the conical layout of $L = 3$, we first used a set of nodes with $K = 25$ that were arranged uniformly on the surface of the cone placed in the area $[-1,1] \times [-1,1] \times [-1,1]$. The cone was sliced into 4 sections of the same height along the vertical axis. Then, 1 (vertex), 4, 8, and 12 (bottom) nodes were uniformly placed on the outer part of the 4 cut surfaces. In the following step, the number of slices was increased by 7, and 1 (vertex), 4, 8, 12, 16, 20, and 24 (bottom) nodes were uniformly arranged in the same way. In both cases, we set $\boldsymbol{\xi}_g = \boldsymbol{\xi}_r = (1/3,3)$, the number of iterations in MCMC was set to $T = 10{,}000$ with the burn-in step $T_b = 5{,}000$, the number of iterations in the third step of fine-tuning was set to $T = 10$, and PTG was run 10 times under the same hyperparameter settings mentioned above.

To quantitatively evaluate the quality of the periodic tables obtained by PTG with the same hyperparameter settings and different trials, we considered using a table as an element descriptor in machine learning tasks. The modeling procedure and the data set that was used is the same as the one mentioned in Section 3.3.3. We performed five-fold cross-validation on the 12,373 samples for the obtained 10 periodic tables. The prediction errors for the 10 periodic tables are shown in Fig. 3.15 for the square table and Fig. 3.16 for the conical table. As shown in Fig. 3.15, the 10th square periodic table gave the lowest MAE (0.533 eV/atom) out of 10 tables. Therefore, this table was chosen as the final visualization result of the square PTG table, and it corresponds to that shown in Fig. 3.5. Similarly, as shown in Fig. 3.16, the 4th conical periodic table giving the lowest MAE (0.464 eV/atom) was chosen as the final visualization result of the conical PTG table, and it corresponds to that shown in Fig. 3.6.

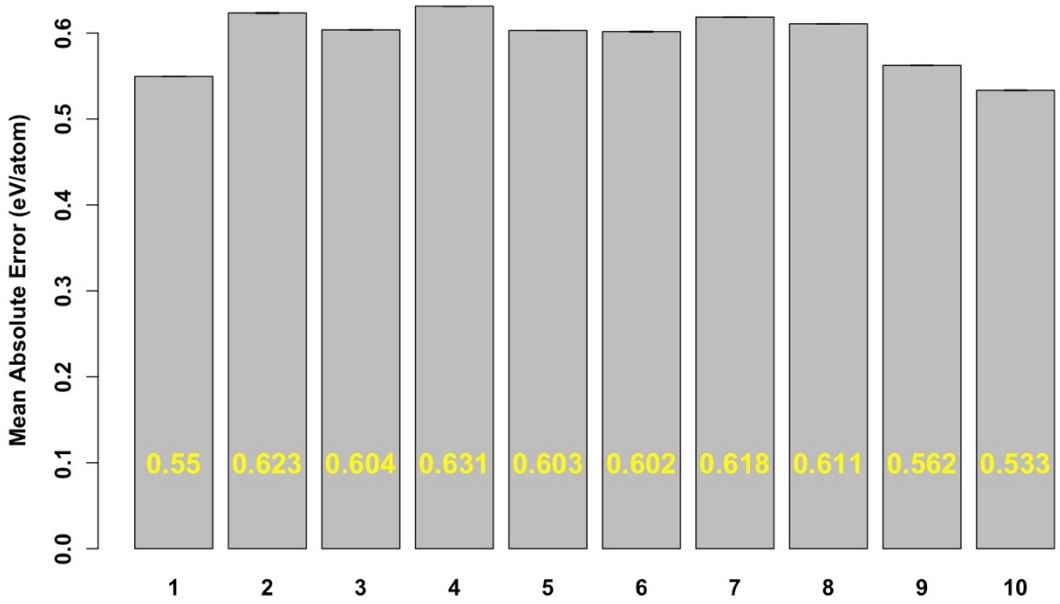


Figure 3.15. Mean absolute errors (MAE) of the prediction of the formation energy per atom for the 10 square periodic tables used as element descriptors. The vertical axis indicates cross-validated MAE of random forest regressors (RF) trained with the 10 descriptors obtained from the coordinates of elements in the square periodic tables produced by PTG, with the same hyperparameters and different trials. The error bars denote the standard deviations in 5 independent trials of the cross-validation.

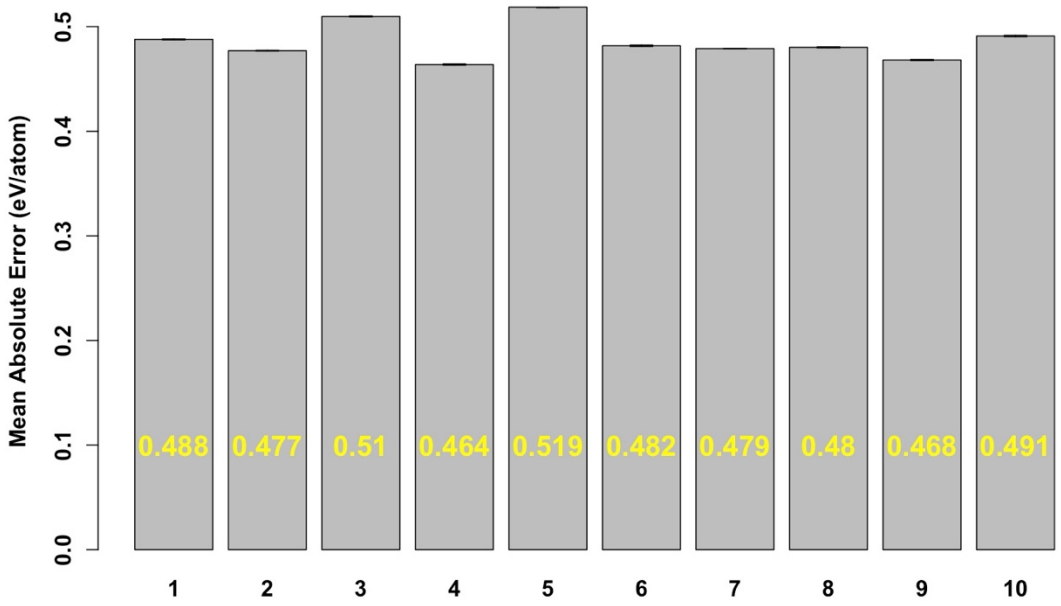


Figure 3.16. Mean absolute errors (MAE) of the prediction of the formation energy per atom for the 10 conical periodic tables used as element descriptors. The vertical axis indicates cross-validated MAE of random forest regressors (RF) trained with the 10 descriptors obtained from the coordinates of elements in the conical periodic tables produced by PTG, with the same hyperparameters and different trials. The error bars denote the standard deviations in 5 independent trials of the cross-validation.

3.7 Other examples

Additionally, we performed PTG on three other node layouts: rectangle, three-dimensional cylinder, and cubic layouts. In the rectangle layout of $L = 2$, we set $K = 27$ in the first step in which the 3×9 nodes were evenly arranged on the area $[-1, 1] \times [-1, 1]$. In the second step, we increased the number of nodes to 5×17 by placing new nodes at the middle points of the line segments connecting between each node. Then, finally, in order to have the same layout as the standard periodic table (5×18), we added a column of 5 nodes in the positive direction of the x-axis. In the cylinder layout of $L = 3$, we first used a set of nodes with $K = 24$ that were arranged uniformly on the surface of the cylinder placed in the area $[-1, 1] \times [-1, 1] \times [-1, 1]$. The cylinder was sliced into 3 sections at the same height along the vertical axis. Then, 8 nodes were uniformly placed on the outer part of the 3 cut surfaces. In the next step, the number of slices was increased by 5, and 16 nodes were uniformly arranged in the same way. In the cubic layout of $L = 3$, we set $K = 27$ in the first step in which the $3 \times 3 \times 3$ nodes were evenly arranged on the area $[-1, 1] \times [-1, 1] \times [-1, 1]$. In the second step, we increased the number of nodes to $5 \times 5 \times 5$ by placing new nodes at the middle points of the line segments connecting between each node. In all the three cases, the element data, the conditions of hyperparameters, and the analysis procedure are completely the same as in the square and conical cases. The results are shown in Fig. 3.17.

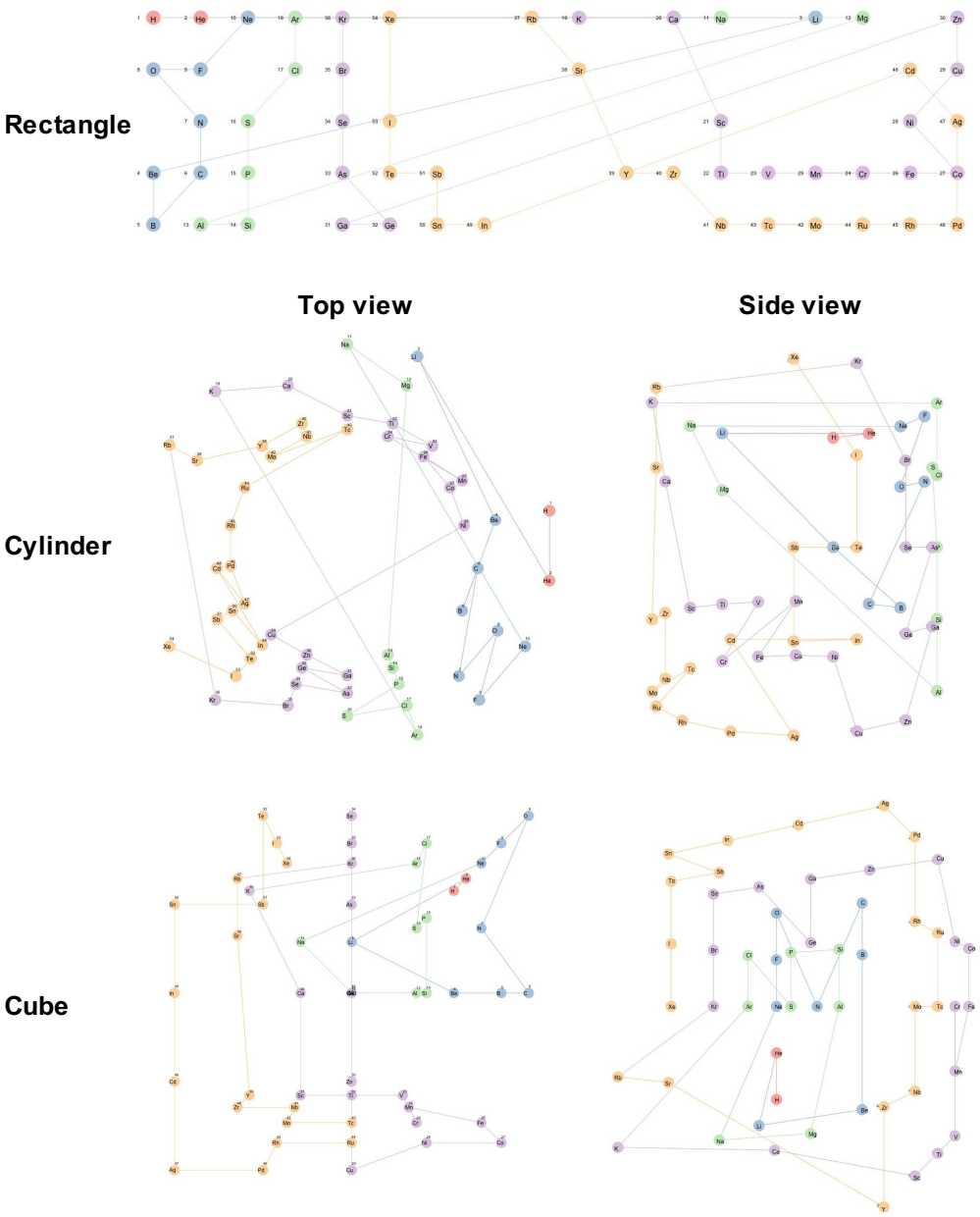


Figure 3.17. Examples of the PTG tables with three different layouts: rectangular grids (top), three-dimensional cylinder type (middle), and cubic (bottom). The elements are color-coded according to the five periods and numbered by atomic numbers. A line passing through the elements is drawn in the order of the atomic numbers. For the cylinder and the cubic tables, the left and right figures show the views from the stop and side, respectively.

3.8 Concluding remarks

Since the emergence of Mendeleev's periodic table, hundreds of redesigned tables have been created. In terms of machine learning, the tabular construction can be considered a task of reducing the dimensionality of high-dimensional data. A previous study first attempted to yield the periodic table using machine learning by applying SOM to five element features available in Mendeleev's time [99]. Although SOM successfully placed similarly behaved elements in neighboring sub-regions on the table, the reported results still never reached Mendeleev's achievement as it obviously failed to capture the underlying periodicity of the elements. With this in mind, we attempted to develop PTG as an unsupervised machine learning algorithm that can automate the translation of high-dimensional data into a tabular form with varying layouts on-demand. The proposed method is applicable as long as a feature set and a template of the table are given. The task of compiling data into tabular displays is the most basic task in data analysis. Nonetheless, there has been considerably less research of this kind in data science so far.

In the previous study based on SOM, some chemical elements with similar properties occupied the same cell in the table due to SOM's inability to guarantee non-overlapping assignments of elements. When we began this study, there were no existing machine learning methods for tabular construction. To the best of our knowledge, the PTG algorithm that we presented is the first tabular constructor based on machine learning, yet this is a secondary contribution of this study.

In this study, we created two types of periodic tables with three additional layouts. The square table was considerably similar to the currently most common periodic table but some outstanding differences were observed; for example, in the arrangement of H and He. These elements were placed far away in the standard periodic table but their physicochemical properties were similar. PTG suggested that these elements should be put closer according to the observed data. The three-dimensional layout on the cone also provided some insight into how the transition elements in the fourth period, including aluminum (Al), should be arranged. In addition, the created conical table provided a re-ordering from Cr to Mn in period 4 and from Mo to Tc in period 5 in the standard table. The results of the intrinsic dimension estimation on the element data showed that the intrinsic dimension of the data is larger than two dimensions (it was estimated as 3–4), at least locally. This result suggests that the element features cannot be fully captured by a two-dimensional representation such as the standard periodic table. Furthermore, the analysis on the point-wise intrinsic dimension of the element data suggested that the tables produced by PTG are clustering the elements according to their local intrinsic dimensions.

A periodic table is the most basic descriptor of chemical elements. Historically, the primary design objective has focused on the understandability and interpretability to humans even at the expense of reducing some key detailed features. Here, we provided a new way of looking at periodic tables. The coordinates of elements put on a table can be considered as an element descriptor, which can also be converted to a descriptor of materials. The quality of designed tables should be assessed on the performance of predicting physicochemical properties of resulting machine learning models. This study focused only on the prediction of formation energies but more diverse properties should be incorporated into the design objective. In addition, we focused only on two types of layouts but there are many more potentially promising options available. Our algorithm would contribute to the creation of more sophisticated tabular displays of chemical elements.

4 Crystal structure prediction using machine learning-based element substitution

The prediction of energetically stable crystal structures formed by a given chemical composition is a central problem in solid-state physics. In principle, the crystalline state of assembled atoms can be determined by optimizing the energy surface, which in turn can be evaluated using first-principles calculations. However, the iterative gradient descent on the potential energy surface using first-principles calculations is prohibitively expensive for complex systems, such as those with many atoms per unit cell. Here, we present a unique methodology for crystal structure prediction (CSP) that relies on a machine learning algorithm called metric learning. It is shown that a binary classifier, trained on a large number of already identified crystal structures, can determine the isomorphism of crystal structures formed by two given chemical compositions with an accuracy of approximately 96.4%. For a given query composition with an unknown crystal structure, the model is used to automatically select from a crystal structure database a set of template crystals with nearly identical stable structures to which element substitution is to be applied. Apart from the local relaxation calculation of the identified templates, the proposed method does not use ab initio calculations. The potential of this substation-based CSP is demonstrated for a wide variety of crystal systems.

4.1 Introduction

Here, we present a powerful CSP method based on machine-learned element substitution. As explained earlier, the method relies on a machine learning algorithm referred to as metric learning [10]. This algorithm is used to automate the selection of template structures from a crystal structure database with high chemical replaceability to the unknown stable structure for a given chemical composition. In metric learning, a binary classifier is constructed to determine whether the crystal structures of two given chemical compositions are identical or not. Crystals with sufficiently high structural similarity are treated as identical, and the labeled dataset is extracted from the crystal structure database. The prediction accuracy of the trained model exceeds 96.4%. Solving the inverse problem of the trained classifier by performing a thorough screening over a large number of known crystals, a set of compositions—as well as their crystal structures that are highly replaceable to a given query composition—can be identified. Then, a template structure is created by assigning the constituent elements in the query composition to the selected template, and a stable crystalline form is obtained by relaxing the created template structure to reach the local minimum energy using DFT calculations. The existing substitution-based methods described earlier statistically estimate the replaceability of two chemical elements based on the observed frequency of their occurrence in two similar crystal structures. As a result, co-occurrence patterns with other elements are completely ignored. Another problem is that, in principle, the model cannot recognize what is dissimilar because previous methods do not use any data on non-identical structures during model training. The proposed method improves the prediction accuracy and extends the applicability domain of the model by learning the replaceability of the overall context of chemical compositions, rather than a pair of elements, with training instances from both similar and dissimilar structures. We show that, in estimation, our substitution-based approach can predict stable structures of approximately 50% of all crystals discovered so far with high confidence. The code for the CSP method is available at [112].

4.2 Method

4.2.1 Outline

Let C_i be a chemical composition and S_i be the corresponding stable crystal structure. The chemical composition C_i is characterized by a descriptor vector $\boldsymbol{\phi}(C_i) \in \mathbb{R}^d$ that encodes d features of the constituent elements in C_i, as detailed below. For a given pair of chemical compositions C_i and C_j, we assign a binary class label y_{ij}, which takes the value 1 if the corresponding stable structures S_i and S_i are significantly close, and 0 otherwise. Here, we construct a model f that predicts the structural similarity label y_{ij} for any given pair of C_i and C_j. The model learns via the supervision of known crystals and their compositions in a crystal structure database. The model takes $\boldsymbol{\phi}(C_i)$ and $\boldsymbol{\phi}(C_j)$ as inputs and it outputs a classification probability f representing their structural identity, which serves as a metric for structural similarity or replaceability between C_i and C_j. The problem thus reduces to the task of metric learning [10].

The trained model (metric) f was used for CSP. For a query composition C_q, our goal is to predict the stable structure (denoted by S_q). Let us assume that the database records N chemical compositions $C_1, \dots, C_N$ and their stable structures $S_1, \dots, S_N$. If the database contains crystal structures that are sufficiently close to S_q, CSP can be performed by screening out those crystals. We can then evaluate the structural similarity between S_q and S_i $(i = 1, \dots, N)$ by assigning $\boldsymbol{\phi}(C_q)$ and $\boldsymbol{\phi}(C_i)$ to f, and selecting the top--K structures as templates. The element species in C_q are assigned to the atoms in each of the top-K selected structures, which are assigned to the DFT calculations to fine-tune the atomic configuration to decrease the free energy. The workflow is summarized in Fig. 4.1.

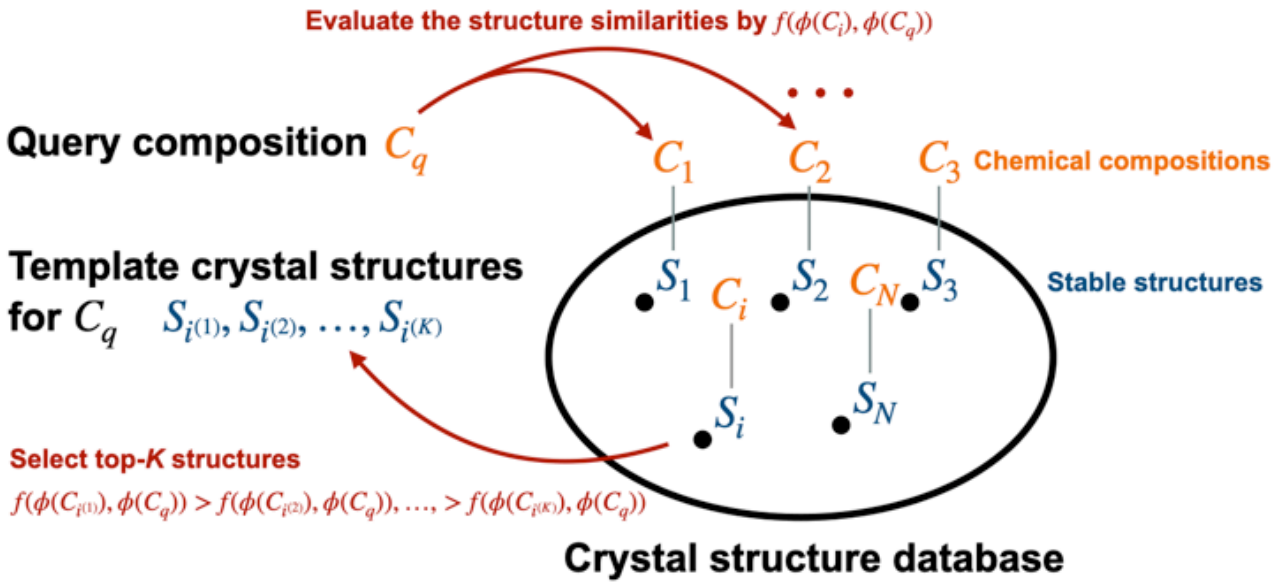


Figure 4.1. Schematic depiction of the substitution-based CSP using metric learning.

4.2.2 Learning to predict structural identity from compositional features

From the given candidate compounds, we select the crystal structures that are predicted to be similar to S_q using a metric f that represents structural similarity.

The model f is constructed using a metric learning algorithm. A training dataset is prepared by taking compound pairs $\{(C_i, C_j, y_{ij})|i = 1, \ldots, M\}$ from the crystal structure database. A structural similarity label is assigned to each (C_i, C_j) by applying a threshold value $\tau = 0.3$ to the crystal structure similarity measure calculated using the local structure order parameters [37] (see Section 4.2.5 for details). The model describes the probability of classifying the structural identity as a function of $\{\boldsymbol{\phi}(C_i), \boldsymbol{\phi}(C_j)\}$. Of the various metric learning methods proposed so far [113, 114], we applied the Siamese network [115] and KISS (keep it simple and straightforward!) metric learning [116], a naïve binary classifier, and a regression model that regresses the structural similarity value instead of y_{ij}. The binary classifier and the regressor were modeled as a conventional multi-layer perceptron (MLP) wherein the input variable is given by the absolute difference between two compositional descriptors, $|\boldsymbol{\phi}(C_i) - \boldsymbol{\phi}(C_j)|$. By comparing the generalization performance of the four metric learning methods mentioned above, we found that binary classification using MLP outperformed others, as shown in Fig. 4.2. Therefore, hereafter, we report only the results of the CSP using a binary classification neural network.

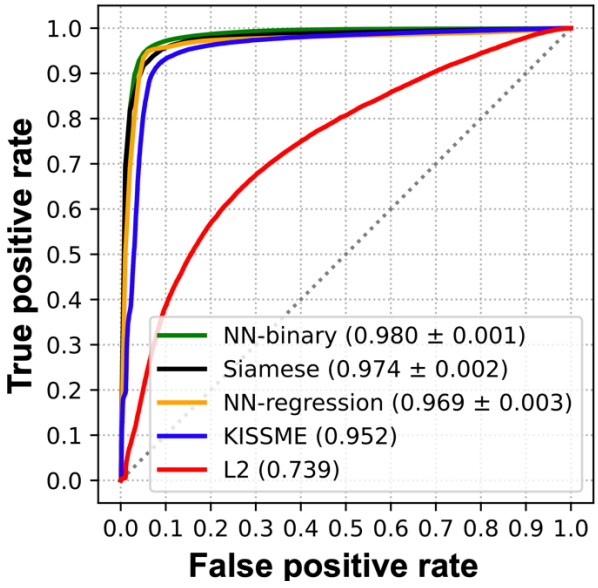


Figure 4.2. Schema receiver operator characteristic (ROC) curves for NN-binary, Siamese network, NN-regression, KISSME, and L2 with mean area under the curve (AUC) and standard deviation of AUC. L2 shows the result of using the Euclidean distance $\|\phi(C_i) - \phi(C_j)\|_2$ as a metric for comparison.

4.2.3 Overall prediction scheme of the CSP method

The overall scheme of the CSP method consists of three steps, as shown in Fig. 4.3. The details of each step are described below.

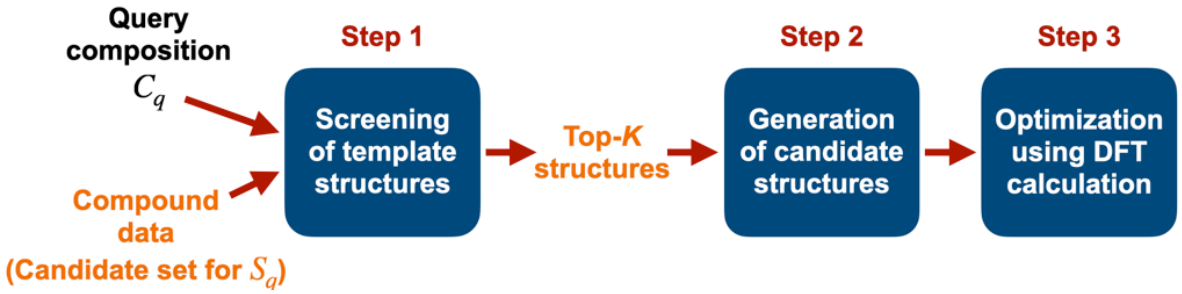


Figure 4.3. Overall scheme of the CSP method.

Step 1: High-throughput screening of template crystal structures

For the CSP of a query composition C_q, the screened candidates are limited to crystals with the same compositional ratio. For example, if Li_3PS_4 is given as the C_q, only crystal structures with a composition ratio of 3:1:4 (the order does not matter) are used as the candidate templates. In the applications shown below, the number of structures to be screened varies from 1 to 3,895 (see Fig. 4.6). The stable structures with the top-K chemical compositions judged to have the highest output probability or structural similarity are selected as the template structure for the query composition.

Step 2: Generation of candidate crystal structures

A crystal structure is created by assigning the constituent elements of the query composition to the atomic coordinates of each selected template. Elements with the same composition ratio between the template and the query are assigned. It is important to note that when one or more elements have the same content ratio, the assignment is not uniquely determined. For example, there are two possible combinations in the assignment of A_1B_1 to C_1D_1 as {A→C & B→D} and {A→D & B→C}. In such cases, a pair of elements with the most similar physicochemical properties should be replaced. Specifically, the element similarity is defined as the Euclidean distance of the 19 elemental descriptors (Table 4.1). The crystal structure generated

inherits the lattice parameters and atomic coordinates of the template structure.

Table 4.1. Detailed description of the 19 elemental descriptors used in step 2.

Feature	Description
atomic_number	Number of protons found in the nucleus of an atom
atomic_weight	The mass of an atom
covalent_radius_pyykko	Single bond covalent radius by Pyykko et al
electron_negativity	Tendency of an atom to attract a shared pair of electrons
en_ghosh	Ghosh's scale of electronegativity
num_unfilled	Total unfilled electron
num_valence	Total valence electron
num_d_unfilled	Unfilled electron in d shell
num_d_valence	Valence electron in d shell
num_f_unfilled	Unfilled electron in f shell
num_f_valence	Valence electron in f shell
num_p_unfilled	Unfilled electron in p shell
num_p_valence	Valence electron in p shell
num_s_unfilled	Unfilled electron in s shell
num_s_valence	Valence electron in s shell
period	Period in the periodic table
group	Group in the periodic table
vdw_radius	Van der Waals radius
vdw_radius_uff	Van der Waals radius from the UFF

Step 3: Geometry optimization using the DFT calculation

Finally, the K candidate structures of C_q are locally optimized by performing DFT calculations. The calculations were performed using the Vienna ab initio Simulation Package (VASP, version 6.1.2) [117], combined with the projector augmented wave pseudopotentials [118]. The exchange-correlation functional was considered with the generalized gradient approximation based on the Perdew-Burke-Ernzerhof method [119]. The Brillouin zone integration for unit cells was automatically determined using the Γ-centered Monkhorst-Pack meshes function implemented in the VASP code. To generate the inputs of VASP calculations, the "MPStaticSet" and "MPRelaxSet" pre-sets implemented in pymatgen [47] were used.

4.2.4 Chemical composition descriptor

We calculated the compositional descriptor, $\boldsymbol{\phi}(C_i) \in \mathbb{R}^d$, using XenonPy [35, 36]. As described earlier, XenonPy is an open-source Python library for materials informatics that we developed, and it provides 58 physicochemical features for each element (Table 4.2). For each element-level feature, the compositional descriptor is calculated by taking summary statistics of constituent elements with the composition ratios such as the weighted mean, weighted sum, weighted variance, and min- and max-pooling. Thus, for a given chemical composition, a 290-dimensional (58 × 5) descriptor vector is defined.

Table 4.2. Detailed description of the 58 elemental descriptors used in the XenonPy [32, 33] descriptor.

Element-level properties used for XenonPy-descriptor

Feature	Description
atomic_number	Number of protons found in the nucleus of an atom
atomic_radius	Atomic radius
atomic_radius_rahm	Atomic radius by Rahm et al
atomic_volume	Atomic volume
atomic_weight	The mass of an atom
boiling_point	Boiling temperature
bulk_modulus	Bulk modulus
c6_gb	C_6 dispersion coefficient in a.u
covalent_radius_cordero	Covalent radius by Cerdero et al
covalent_radius_pyykko	Single bond covalent radius by Pyykko et al
covalent_radius_pyykko_double	Double bond covalent radius by Pyykko et al
covalent_radius_pyykko_triple	Triple bond covalent radius by Pyykko et al
covalent_radius_slater	Covalent radius by Slater
density	Density at 295K
dipole_polarizability	Dipole polarizability
electron_negativity	Tendency of an atom to attract a shared pair of electrons
electron_affinity	Electron affinity
en_allen	Allen's scale of electronegativity
en_ghosh	Ghosh's scale of electronegativity
en_pauling	Pauling's scale of electronegativity
first_ion_en	First ionisation energy
fusion_enthalpy	Fusion heat
gs_bandgap	DFT bandgap energy of T=0K ground state
gs_energy	DFT energy per atom (raw VASP value) of T=0K ground state
gs_est_bcc_latcnt	Estimated BCC lattice parameter based on the DFT volume
gs_est_fcc_latcnt	Estimated FCC lattice parameter based on the DFT volume
gs_mag_moment	DFT magnetic momenet of T=0K ground state
gs_volume_per	DFT volume per atom of T=0K ground state
hhi_p	Herfindahl–Hirschman Index (HHI) production values
hhi_r	Herfindahl–Hirschman Index (HHI) reserves values
heat_capacity_mass	Mass specific heat capacity
heat_capacity_molar	Molar specific heat capacity
icsd_volume	Atom volume in ICSD database
evaporation_heat	Evaporation heat
heat_of_formation	Heat of formation
lattice_constant	Physical dimension of unit cells in a crystal lattice
mendeleev_number	Atom number in mendeleev's periodic table
melting_point	Melting point
molar_volume	Molar volume
num_unfilled	Total unfilled electron
num_valence	Total valence electron
num_d_unfilled	Unfilled electron in d shell
num_d_valence	Valence electron in d shell
num_f_unfilled	Unfilled electron in f shell
num_f_valence	Valance electron in f shell
num_p_unfilled	Unfilled electron in p shell
num_p_valence	Valence electron in p shell
num_s_unfilled	Unfilled electron in s shell
num_s_valence	Valence electron in s shell
period	Period in the periodic table
specific_heat	Specific heat at 20oC
thermal_conductivity	Thermal conductivity at 25 C
vdw_radius	Van der Waals radius
vdw_radius_alvarez	Van der Waals radius according to Alvarez
vdw_radius_mm3	Van der Waals radius from the MM3 FF
vdw_radius_uff	Van der Waals radius from the UFF
sound_velociy	Speed of sound
Polarizability	Ability to form instantaneous dipoles

4.2.5 Preparation of structural similarity labels

Metric learning relies on the supervision of the binary class label y_{ij}, which indicates whether a pair of crystal structures are similar or dissimilar. The class label is calculated as follows: (1) quantify the crystal structure similarities of all compound pairs, and (2) binarize the similarity measures by applying a prescribed threshold τ.

To calculate the structural similarity, we encoded a given structure using the site fingerprint with local structure order parameters [37] (implemented in matminer [48, 49], an open-source toolkit for materials data mining). By evaluating the degree of resemblance of the coordination environment of an atomic site to the preset-coordination motifs, we obtained a vector-type descriptor (site fingerprint) for each atomic site in the crystal structure. Then, a crystal structure descriptor was calculated by taking the summary statistics of the site fingerprints across all atomic sites in the crystal structure. We used the mean, standard deviation, minimum, and maximum as the summary statistics. Finally, the structural similarity was calculated as the Euclidean distance between the crystal structure descriptors. The similarity measure uses only the topological features of the atomic coordinates and does not use any information about the elemental composition.

Following the procedure described above, we calculated 549,544,128 crystal structure dissimilarities between all pairs of the 33,153 stable compounds in the Materials Project database [15, 26] (version released on 11/21/2020). A histogram of dissimilarities is shown in Fig. 4.4. We considered an appropriate threshold for the binarization of similarity measures. The trade-off in the occurrence of false positives and false negatives should be considered when choosing the threshold value: a large threshold value increases the number of false negative cases where structurally similar structures are judged to be dissimilar; a small threshold increases the number of false positives. If the threshold is too small, the number of positive instances (structurally similar pairs) becomes very small, making the treatment of imbalanced data difficult.

To determine the value of τ that appropriately balances the trade-off, we tested the binarization by varying $\tau \in \{0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7\}$. For each τ, we examined the number of instances classified as "similar" and the proportion of compounds that appeared at least once in the class "similar" (Table 4.3). At $\tau = 0.3$, approximately 80% of all the compounds appeared at least once in the class "similar". The remaining 20% were judged to have no similar pairs. This means that the structures of these 20% cannot be predicted using the substitution-based method. In contrast, stable structures of 80% of the crystals can be determined using the substitution-based method. Based on these considerations, we set the threshold to $\tau = 0.3$.

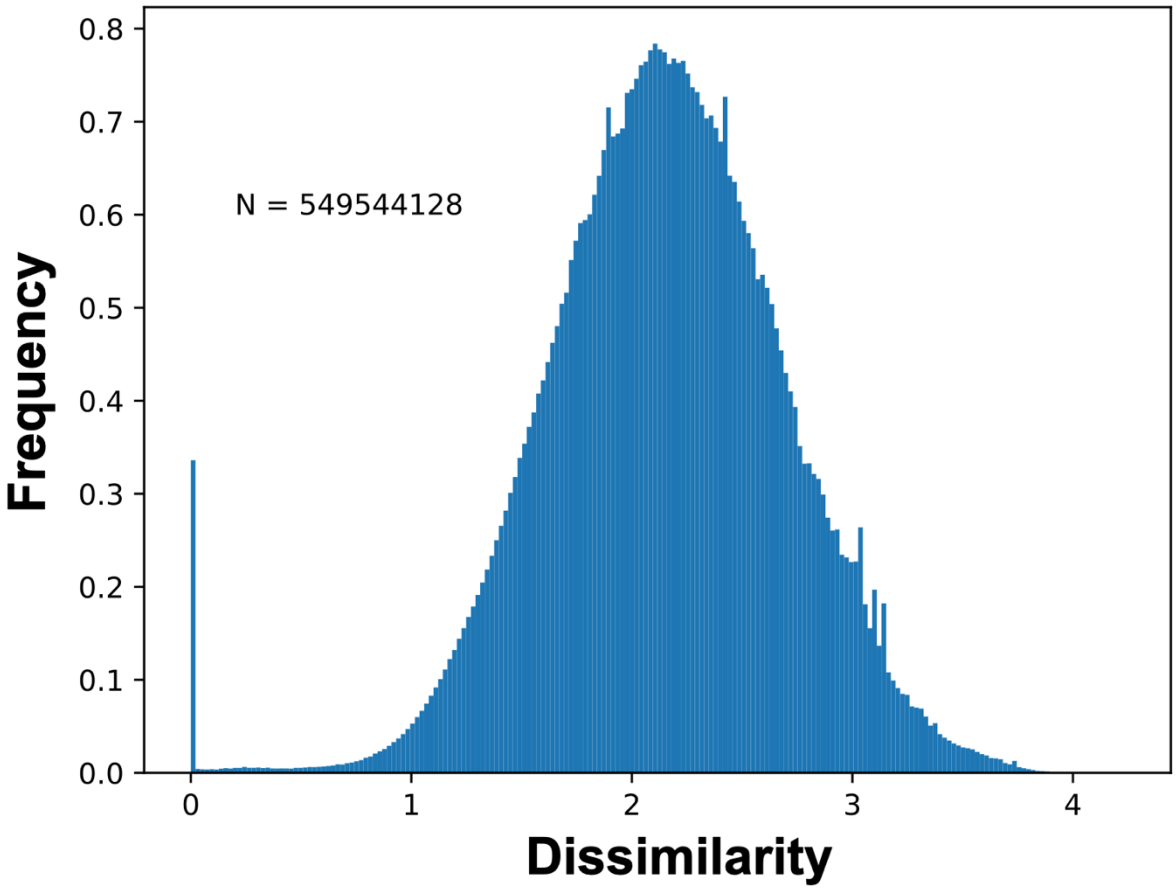


Figure 4.4. Histogram of the structural dissimilarities (the Euclidian distance of the structure fingerprint calculated with local

structure order parameters [37]) of 549,544,128 pairs (all pairs for the 33,153 compounds).

Table 4.3. Results of binarization with varying τ. The first column denotes threshold τ for the structure dissimilarity. The second column denotes the number of compounds that have at least one similar pair in the all candidates (for each compound, the all candidates equal the remaining 33,152 compounds). Here, the definition of the similarity is determined by τ (i.e., similar pair is the pair in which the structure dissimilarity is less than τ). The third column denotes number of similar pairs in the all pairs of the 33,153 stable compounds for each τ.

τ	Number of compounds that have similar pairs	Number of similar pairs
0.01	5,889/ 33,153 (17.76%)	3,886,505/ 549,544,128 (0.71%)
0.1	17,413/ 33,153 (52.52%)	4,077,592/ 549,544,128 (0.74%)
0.2	23,169/ 33,153 (69.89%)	4,320,784/ 549,544,128 (0.79%)
0.3	26,106/ 33,153 (78.74%)	4,628,444/ 549,544,128 (0.84%)
0.4	27,810/ 33,153 (83.88%)	4,912,232/ 549,544,128 (0.89%)
0.5	29,355/ 33,153 (88.54%)	5,178,214/ 549,544,128 (0.94%)
0.6	30,871/ 33,153 (93.12%)	5,518,144/ 549,544,128 (1.0%)
0.7	31,947/ 33,153 (96.36%)	5,967,201/ 549,544,128 (1.09%)

4.2.6 Experimental procedure

From the 126,335 inorganic compounds in the Materials Project database, we obtained 33,153 stable compounds with an energy above the hull equal to zero. To benchmark the predictive performance of the proposed CSP, 38 crystals were selected, taking into account the diversity of space groups, structures, constituent elements, the number of atoms per unit cell, and their application domains; the number of atoms per unit cell was distributed in the range of 2 to 104.

The remaining 33,115 compounds, which were not used for the benchmark, were randomly divided into 10,000 for training, 2,000 for validation, and 21,115 for testing in the process of metric learning. Of the 10,000 training compounds (49,995,000 pairs, ${}_{10000}C_2$), 421,000 pairs were categorized as similar at $\tau = 0.3$. To eliminate the imbalance between the number of positive and negative instances in the training of the classifier, 421,000 negative instances were randomly selected from the 49,574,000 dissimilar groups. Following the same procedure, the validation and test sets were selected so that the number of positive and negative instances was equal, resulting in a total of 32,050 and 3,782,728 pairs, respectively.

The model input was defined as the absolute difference between the 290-dimensional compositional descriptors, $\left|\boldsymbol{\phi}(C_i) - \boldsymbol{\phi}(C_j)\right|$, and the output was given by the similarity label y_{ij}. The binary classifier was independently trained five times using randomly selected training and validation datasets. During each training, the hyperparameters were adjusted to provide the highest prediction accuracy for the validation set (see Section 4.5 for details). The ensemble of these five models, $f_1, \ldots, f_5$, was used to produce the predicted class label. The class probability of being classified into similar pairs is given by $\hat{f}\left(\left|\boldsymbol{\phi}(C_i) - \boldsymbol{\phi}(C_j)\right|\right) = \frac{1}{5}\sum_{b=1}^{5} f_b\left(\left|\boldsymbol{\phi}(C_i) - \boldsymbol{\phi}(C_j)\right|\right)$. For the set of candidate templates, we used all 33,115 stable structures except the 38 benchmark query compositions.

For a given query composition, according to the magnitude of the class probability of being classified into similar pairs in which the 33,115 candidate compounds with known crystal structures were screened out, we identified the top five template structures with a probability greater than 0.5. We then constructed candidate crystal structures as described above, which were optimized using the full structural relaxation in DFT.

4.3 Result

The performance of the ensemble prediction that used five different neural networks was measured based on the dataset consisting of the similar and dissimilar pairs of the test 21,115 compounds. The receiver operator characteristic (ROC) curve [120] according to the varying thresholds of the classification probability is shown in Fig. 4.5. The area under the curve (AUC) [120] and the prediction accuracies were 0.991 and 96.4%, respectively.

According to the performance tests shown above, the similarity of the stable structures of the two given chemical compositions can be predicted with a considerably high accuracy. We applied this similarity prediction model to identify the known stable structures of the 38 benchmark crystals. The results are summarized in Table 4.4. The proposed method was applied to select a maximum of five template structures, which were then subjected to element substitution to produce a set of candidate crystal structures. Hereafter, this set is denoted as $\mathcal{C}_5$. As mentioned above, only the structures with the same composition ratio as the query among the total 33,115 stable structures in the Materials Project were considered as candidates during the screening. We denote this set as $\mathcal{C}$. The number of candidate structures selected here varies greatly depending on the query compositions: for all the stable structures in the Materials Project, the mean and median numbers of structures with the same composition ratio were 322.0 and 952.0, respectively, and the maximum and minimum numbers were 3,895 and 0, respectively. By definition, a crystal structure with the absence of matching structures is unpredictable (1,051/33,153) by the proposed method. The distribution of the number of matches is shown in Fig. 4.6. It can be seen that the number of candidate structures is considerably reduced after composition ratio matching.

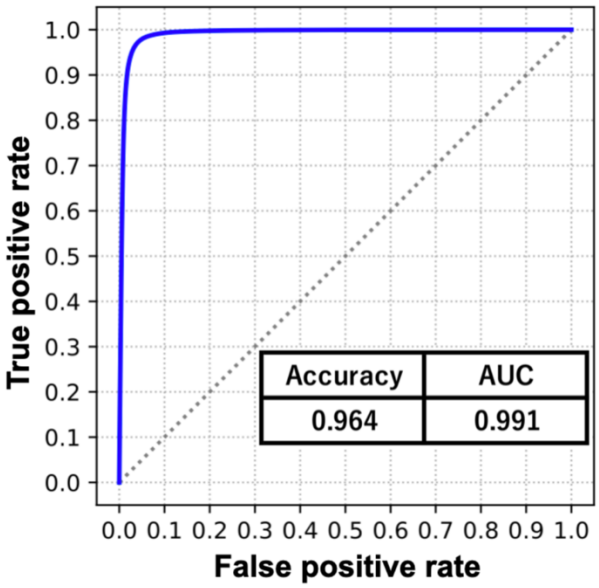


Figure 4.5. ROC curve and performance metrics (accuracy and AUC) in the prediction of the structural identity using fully connected neural networks.

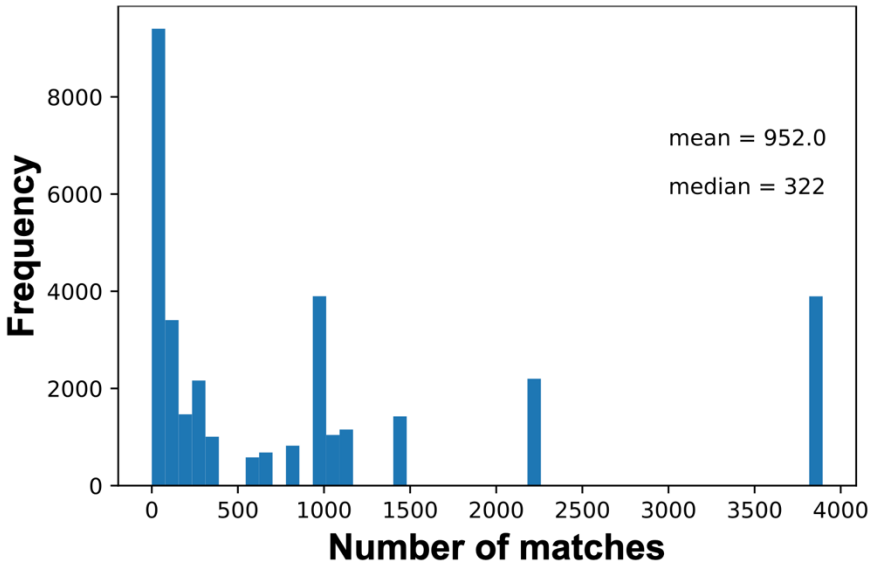


Figure 4.6. Distribution of the number of matches of the composition ratio matching for the 33,153 compounds. The maximum and minimum numbers were 3895 and 0. Here, the number of matches = 0 mean the absence of matching structures (in 1,051

out of 33,153 compounds, the number of matches = 0).

Table 4.4. Results of the CSP for the 38 benchmark systems. The first column lists the 35 query compositions that were predictable. The second column lists the minimum dissimilarities of all candidate structures that have the same composition ratio as the query. The third column presents the minimum structural dissimilarities for the top five identified structures with respect to the true stable structure. The fourth column lists the rank of the minimum dissimilarity of the top five candidates in all candidates. For example, the top five candidates of Al_2O_3 are ranked 2nd out of 297 candidates; thus, the cell is marked 2/297. The fifth column indicates whether the true structure is included in the top five predicted structures relaxed by DFT (✓ and − indicate success and failure, respectively).

Composition	Min. dissimilarity of all candidates	Min. dissimilarity of top 5	Rank	Prediction success
Ag_8GeS_6	0.214	0.214	1/34	–
Al_2O_3	0.067	0.093	2/297	✓
BN	1.726	3.292	683/960	–
$Ba(FeAs)_2$	0.091	0.176	9/1424	✓
Bi_2Te_3	0.293	0.293	1/297	✓
C	1.769	1.975	3/87	–
$Ca_{14}MnSb_{11}$	0.083	0.096	2/13	✓
$CaCO_3$	0.054	0.077	3/1000	✓
Cd_3As_2	0.19	0.19	1/297	✓
$CoSb_3$	0.068	0.068	1/1042	✓
$CsPbI_3$	0.129	0.129	1/1000	✓
$Cu_{12}Sb_4S_{13}$	0.24	0.24	1/1	✓
Fe_3O_4	0.216	0.216	1/152	–
GaAs	0	0	1/960	✓
GeH_4	0.383	0.639	22/171	–
La_2CuO_4	0.022	0.022	1/821	✓
Li_3PS_4	0.851	1.216	33/250	–
$Li_4Ti_5O_{12}$	0.282	0.282	1/8	–
$LiBF_4$	0.302	0.592	6/983	–
$LiCoO_2$	0.199	0.207	5/3895	–
$LiFePO_4$	0.113	0.13	2/327	✓
$LiPF_6$	0.046	0.297	6/242	✓
$Mn(FeO_2)_2$	0.022	0.022	1/821	✓
Si	0	2.304	7/87	–
Si_3N_4	0.269	0.269	1/152	–
SiO_2	0.167	0.167	1/1151	–
$SrTiO_3$	0.395	0.643	16/1000	✓
TiO_2	1.015	1.401	20/1151	–
V_2O_5	0.753	1.865	41/85	–
VO_2	0.077	0.077	1/1151	✓
$Y_3Al_5O_{12}$	0.014	0.014	1/49	✓
ZnO	0.006	0.062	5/960	✓
ZnSb	0.316	0.316	1/960	✓
ZrO_2	0.131	0.131	1/1151	✓
$ZrTe_5$	0.039	0.039	1/132	✓

The CSP method could not propose any template for 3 out of the 38 query compositions, $NaCaAlPHO_5F_2$, MgB_7, and $Ba_2CaSi_4(BO_7)_2$: none of the candidates had the same composition ratio as $NaCaAlPHO_5F_2$ in the 33,115 candidates; for MgB_7 and $Ba_2CaSi_4(BO_7)_2$, none of the candidates had class probabilities greater than 0.5. Table 4.4 summarizes the prediction results for the remaining 35 query compositions. We investigated the dissimilarity value of the closest structure in $\mathcal{C}_5$ to the known stable structure (the third column), which was compared with the minimum dissimilarity value of the best template among all candidates in $\mathcal{C}$ (the second column). The structural dissimilarities were calculated with the local structure order parameter presented in [37]. This measure was defined only on atomic coordinates, ignoring the information of element types. For quantitative evaluation, the rank of the minimum dissimilarity value of the top 5 candidates was calculated with respect to the dissimilarities of $\mathcal{C}$ (the fourth column). The model could select the best template, which is the closest to the true stable structure in $\mathcal{C}$, with an accuracy of approximately 51.4% (=18/35) by screening out the top 5 candidates. Only 2 cases, Li3PS4 and BN, had ranks higher than 30. With the top 5 candidates, the proposed method succeeded in identifying templates that are almost equivalent to the best template.

After the element substitution, the candidate crystal structures were locally optimized using DFT calculations. The predicted and true crystal structures for 12 arbitrarily selected queries are shown in Fig. 4.7. Of the top five, the predicted structure closest to the true structure is illustrated. The predicted crystal structures for all the 35 queries are shown in Fig. 4.8. It can be seen that highly complex crystal structures consisting of a large number of atoms per unit cell could be successfully predicted, thereby demonstrating a definitive improvement over ordinary CSP programs. We determined whether the true structure was included in the top five relaxed predicted structures by performing a visual inspection. As summarized in Table 4.4 (the fifth column), in 21 out of the 35 queries (60%), the top five predicted structures contained the true stable structure.

Furthermore, we examined the key factors that determine success or failure. First, for the failed and successful query compositions, the median, mean, and standard deviation of the number of elements were calculated. The respective values of the statistics were 3.0, 2.57, 0.58 (for success) and 2.0, 2.21, 0.67 (for failure). Thus, there was no significant difference between the successful and unsuccessful cases. For the number of atomic sites per unit cell, the three statistics also showed no significant difference: 20.0, 42.5, 56.3 (for successful queries) and 19.0, 29.1, 23.5 (for unsuccessful queries). In contrast, when the dissimilarity between the top five template structures and the true structure was compared between success and failure, the median, mean, and variance showed remarkable differences: 0.096, 0.148, 0.145 (template structures) and 0.615, 1.0416, 0.954 (true structure). Therefore, in the proposed CSP, the selection of a template structure sufficiently close to the true structure is the dominant factor that determines the success or failure.

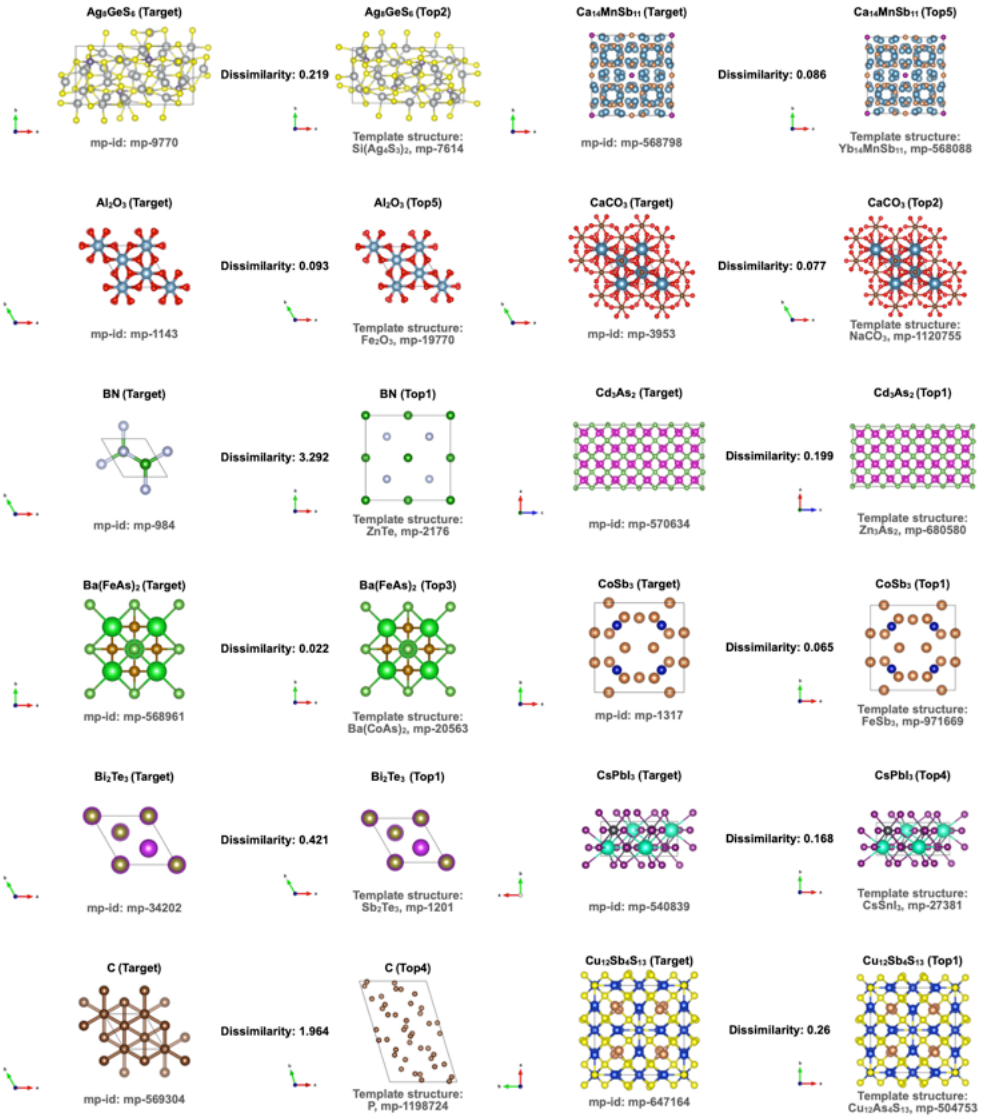


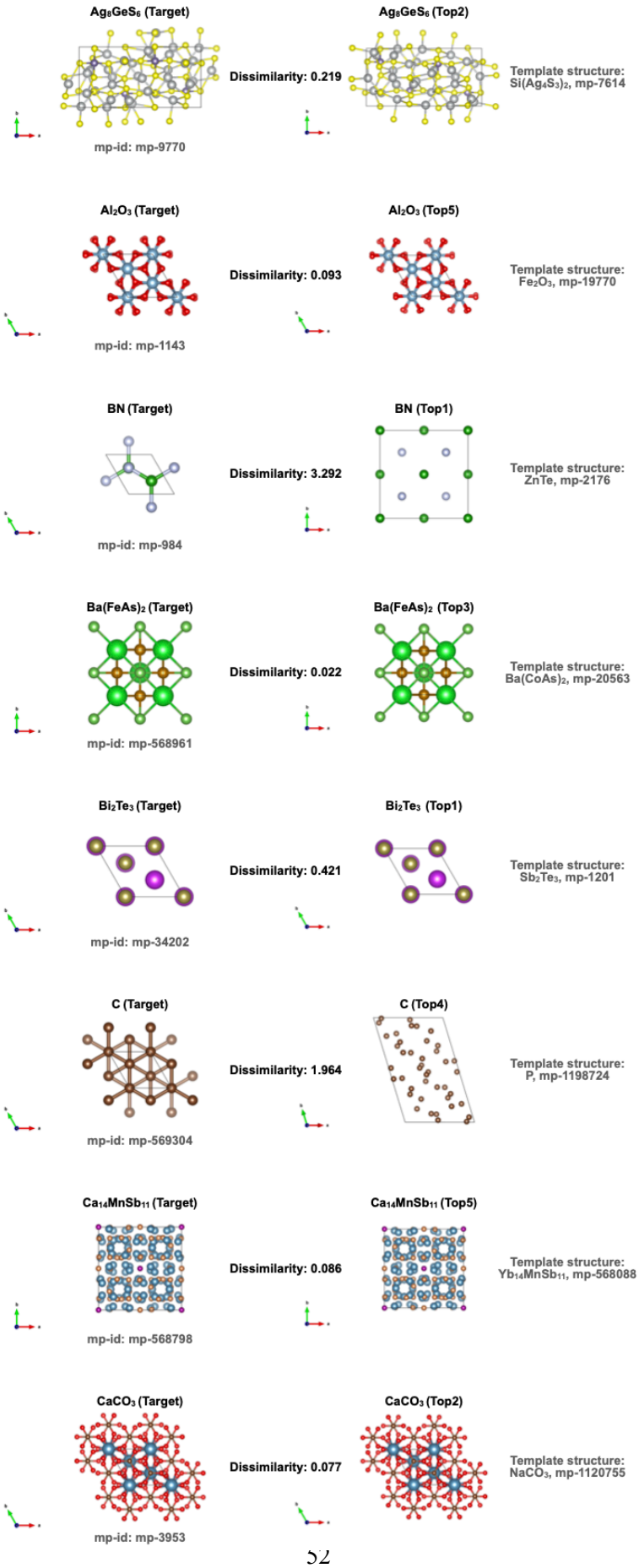
Figure 4.7. 12 examples of the predicted and true crystal structures. The closest predicted structure to the true structure among the top 5 candidates (depicted with VESTA [121]) is shown. For all the results of the 35 test cases, see Fig. 4.8.

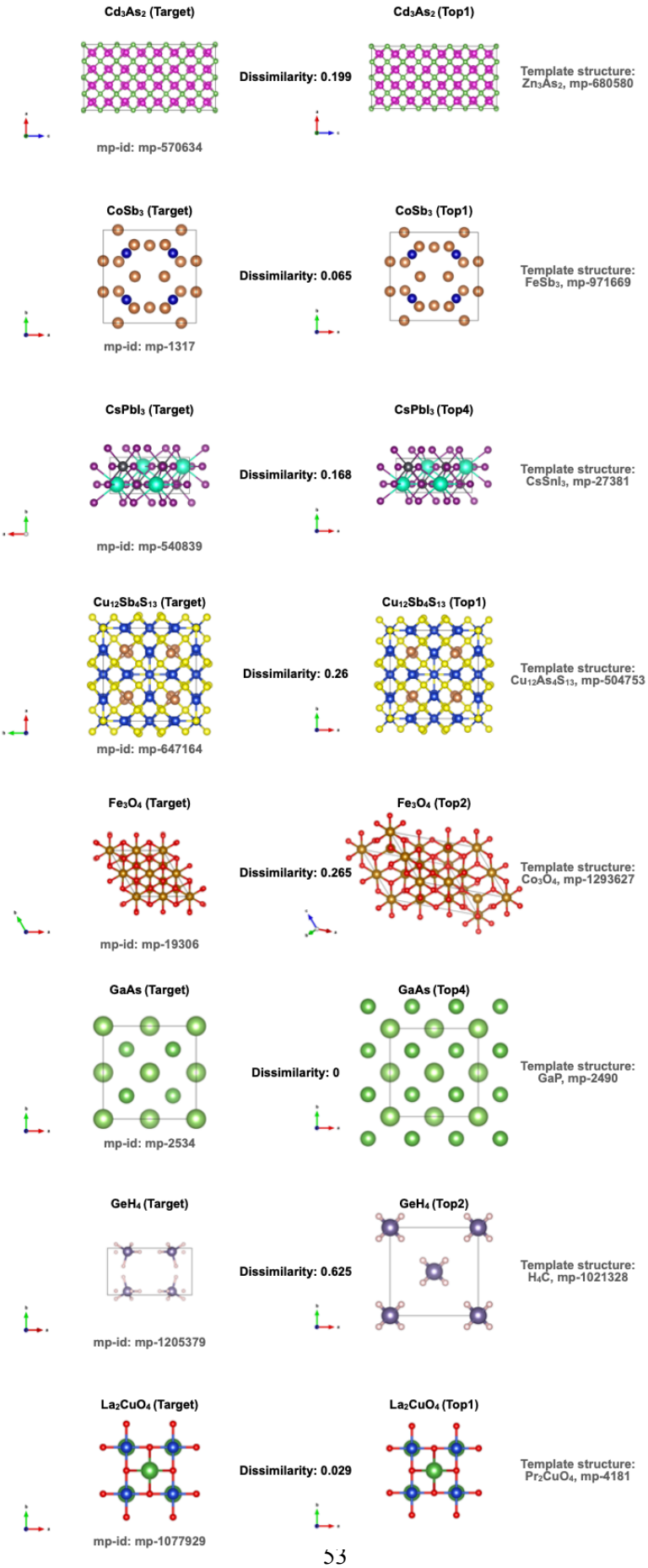
Finally, we estimated the extent to which elemental substitution could cover the entire crystal system in predicting stable crystal structures. Although only 38 benchmark compositions were tested, we investigated the ability to identify the best templates for the 21,115 stable crystals selected randomly from the Materials Project. Each of the 21,115 crystals was used as a query to select a candidate template from the known stable structures. Here, we predicted the stable structure of the query composition without

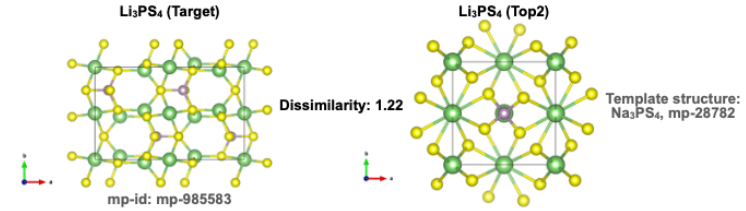
performing DFT structural relaxation (Table 4.5). For the top 50 identified templates, 10,829 out of 21,115 (=51.3%) stable crystals were found to have true stable structures within a radius of 0.1 of the structural dissimilarity. This corresponds to a prediction accuracy of 99.2% (= 10, 829/10, 914). Table 4.5 also summarizes the performance of detecting the best templates with the top K predicted templates when the number of identified templates (K) was varied from 1 to 50, and the radius threshold was varied from 0.1 to 0.3. As shown in Table 4.4, when a template structure with a structural dissimilarity less than 0.1 could be selected, the proposed method could identify the true stable structures with 100% accuracy (= 11/11) by performing structural relaxation with DFT. Therefore, we estimate that approximately 51.3% (51.3 × 1.0) of the entire crystal system can be predicted using the proposed method. Moreover, if, for example, the threshold of the radius was set to 0.2, the proportion of the best templates falling within the given radius of the top 50 candidates was 66.8%, and the structural relaxation using DFT converged to the true stable structures for approximately 94.1% (= 16/17) of the crystals. Therefore, it is estimated that approximately 62.8% (66.8 × 0.94) of the entire crystal system can be predicted by our method.

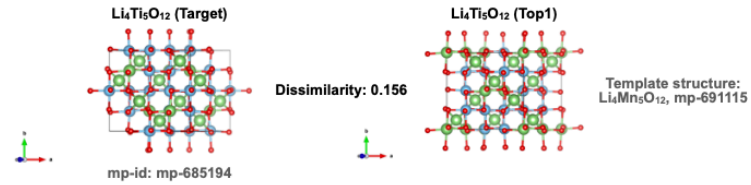
Table 4.5. Results of the best template prediction for 21,115 unique crystal systems in the Materials Project. The first column lists the threshold τ for the dissimilarities. The second column denotes the proportions of systems that had at least one candidate with a dissimilarity of less than τ in all candidates. The rest of the columns denote the proportions of the systems that had at least one candidate with a dissimilarity of less than τ in the top 1, 5, 10, 20, 30, and 50 suggested candidates, respectively.

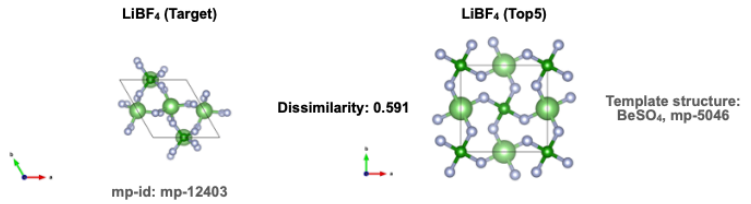
τ	**All candidates**	**Top 1**	**Top 5**	**Top 10**	**Top 20**	**Top 30**	**Top 50**
0.1	51.7%	31.7%	44.8%	47.7%	49.7%	50.5%	51.3%
0.2	68.1%	46.0%	60.7%	63.5%	65.3%	66.0%	66.8%
0.3	76.4%	55.5%	69.8%	72.4%	73.9%	74.5%	75.1%

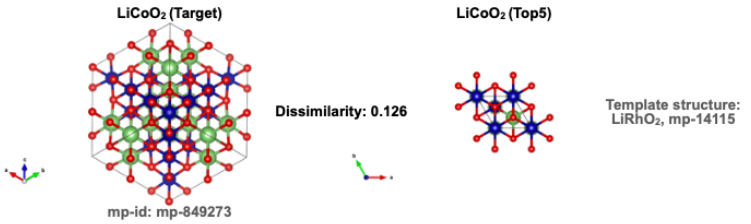

Ag8GeS6 (Target)
mp-id: mp-9770
Dissimilarity: 0.219
Ag8GeS6 (Top2)
Template structure: Si(Ag4S3)2, mp-7614
Al2O3 (Target)
mp-id: mp-1143
Dissimilarity: 0.093
Al2O3 (Top5)
Template structure: Fe2O3, mp-19770
BN (Target)
mp-id: mp-984
Dissimilarity: 3.292
BN (Top1)
Template structure: ZnTe, mp-2176
Ba(FeAs)2 (Target)
mp-id: mp-568961
Dissimilarity: 0.022
Ba(FeAs)2 (Top3)
Template structure: Ba(CoAs)2, mp-20563
Bi2Te3 (Target)
mp-id: mp-34202
Dissimilarity: 0.421
Bi2Te3 (Top1)
Template structure: Sb2Te3, mp-1201
C (Target)
mp-id: mp-569304
Dissimilarity: 1.964
C (Top4)
Template structure: P, mp-1198724
Ca14MnSb11 (Target)
mp-id: mp-568798
Dissimilarity: 0.086
Ca14MnSb11 (Top5)
Template structure: Yb14MnSb11, mp-568088
CaCO3 (Target)
mp-id: mp-3953
Dissimilarity: 0.077
CaCO3 (Top2)
Template structure: NaCO3, mp-1120755

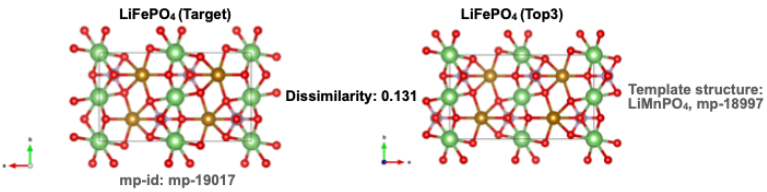

Cd3As2 (Target)
mp-id: mp-570634
Dissimilarity: 0.199
Cd3As2 (Top1)
Template structure: Zn3As2, mp-680580
CoSb3 (Target)
mp-id: mp-1317
Dissimilarity: 0.065
CoSb3 (Top1)
Template structure: FeSb3, mp-971669
CsPbI3 (Target)
mp-id: mp-540839
Dissimilarity: 0.168
CsPbI3 (Top4)
Template structure: CsSnI3, mp-27381
Cu12Sb4S13 (Target)
mp-id: mp-647164
Dissimilarity: 0.26
Cu12Sb4S13 (Top1)
Template structure: Cu12As4S13, mp-504753
Fe3O4 (Target)
mp-id: mp-19306
Dissimilarity: 0.265
Fe3O4 (Top2)
Template structure: Co3O4, mp-1293627
GaAs (Target)
mp-id: mp-2534
Dissimilarity: 0
GaAs (Top4)
Template structure: GaP, mp-2490
GeH4 (Target)
mp-id: mp-1205379
Dissimilarity: 0.625
GeH4 (Top2)
Template structure: H4C, mp-1021328
La2CuO4 (Target)
mp-id: mp-1077929
Dissimilarity: 0.029
La2CuO4 (Top1)
Template structure: Pr2CuO4, mp-4181

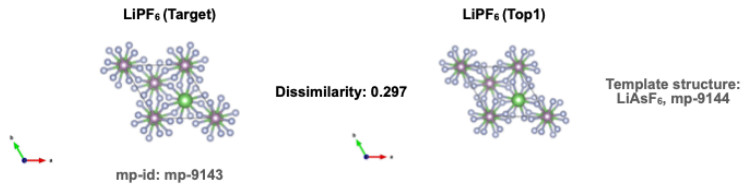

Li3PS4 (Target)
mp-id: mp-985583
Dissimilarity: 1.22
Li3PS4 (Top2)
Template structure:
Na3PS4, mp-28782

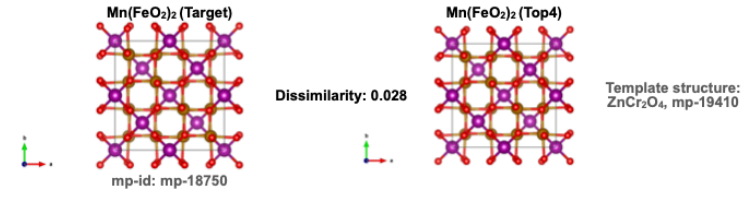

Li4Ti5O12 (Target)
mp-id: mp-685194
Dissimilarity: 0.156
Li4Ti5O12 (Top1)
Template structure:
Li4Mn5O12, mp-691115

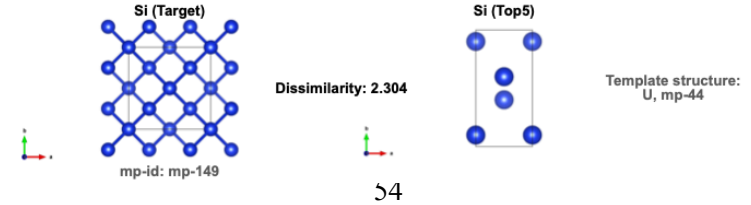

LiBF4 (Target)
mp-id: mp-12403
Dissimilarity: 0.591
LiBF4 (Top5)
Template structure:
BeSO4, mp-5046

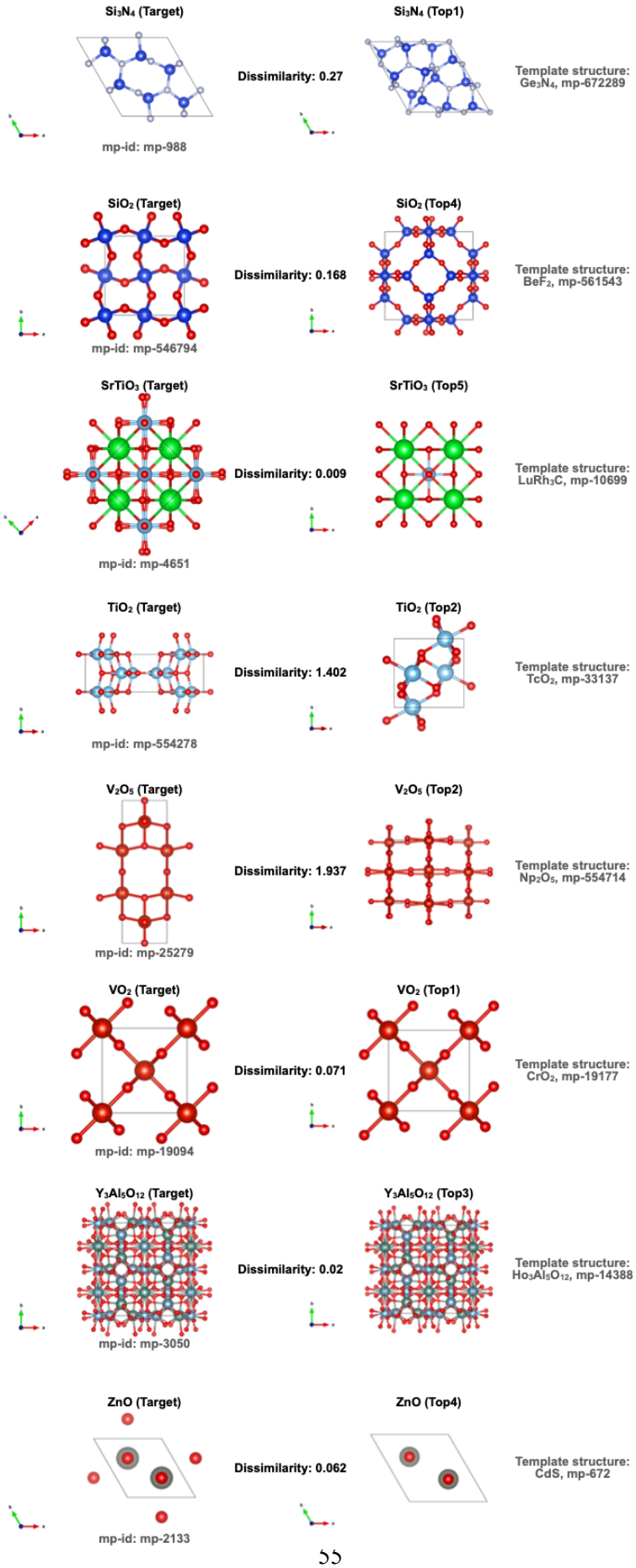

LiCoO2 (Target)
mp-id: mp-849273
Dissimilarity: 0.126
LiCoO2 (Top5)
Template structure:
LiRhO2, mp-14115


LiFePO4 (Target)
mp-id: mp-19017
Dissimilarity: 0.131
LiFePO4 (Top3)
Template structure:
LiMnPO4, mp-18997


LiPF6 (Target)
mp-id: mp-9143
Dissimilarity: 0.297
LiPF6 (Top1)
Template structure:
LiAsF6, mp-9144


Mn(FeO2)2 (Target)
mp-id: mp-18750
Dissimilarity: 0.028
Mn(FeO2)2 (Top4)
Template structure:
ZnCr2O4, mp-19410


Si (Target)
mp-id: mp-149
Dissimilarity: 2.304
Si (Top5)
Template structure:
U, mp-44


Si3N4 (Target)
mp-id: mp-988
Dissimilarity: 0.27
Si3N4 (Top1)
Template structure: Ge3N4, mp-672289
SiO2 (Target)
mp-id: mp-546794
Dissimilarity: 0.168
SiO2 (Top4)
Template structure: BeF2, mp-561543
SrTiO3 (Target)
mp-id: mp-4651
Dissimilarity: 0.009
SrTiO3 (Top5)
Template structure: LuRh3C, mp-10699
TiO2 (Target)
mp-id: mp-554278
Dissimilarity: 1.402
TiO2 (Top2)
Template structure: TcO2, mp-33137
V2O5 (Target)
mp-id: mp-25279
Dissimilarity: 1.937
V2O5 (Top2)
Template structure: Np2O5, mp-554714
VO2 (Target)
mp-id: mp-19094
Dissimilarity: 0.071
VO2 (Top1)
Template structure: CrO2, mp-19177
Y3Al5O12 (Target)
mp-id: mp-3050
Dissimilarity: 0.02
Y3Al5O12 (Top3)
Template structure: Ho3Al5O12, mp-14388
ZnO (Target)
mp-id: mp-2133
Dissimilarity: 0.062
ZnO (Top4)
Template structure: CdS, mp-672

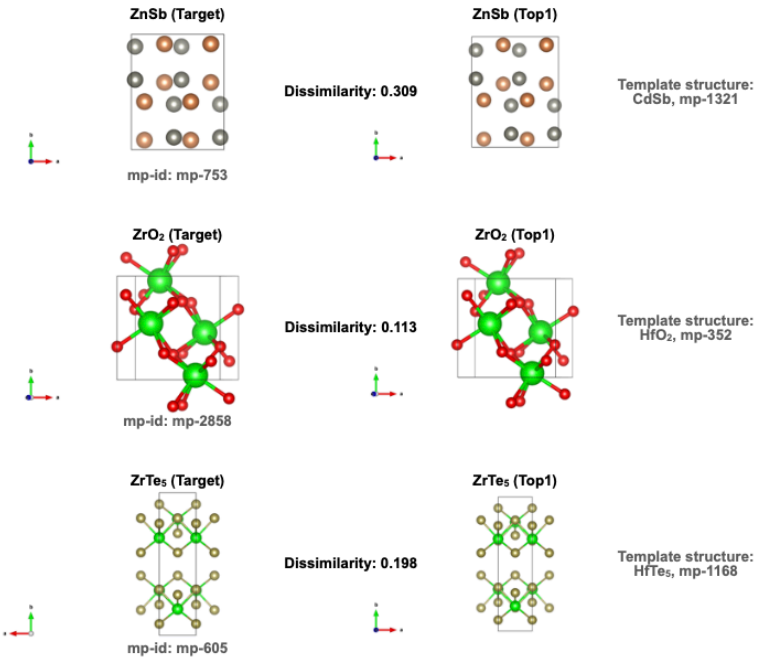


Fig. 4.8. Results of the structure prediction for the 35 query compositions (the order in which they are displayed is the same as the first column of Table 4.4). The crystal structure diagrams of the true structures, and the predicted structures (the closest one to the true structure in top 5 structures) for each query are shown here (depicted with VESTA [121]). The dissimilarity values between the true structure and the predicted structure, which were calculated with local structure order parameters [37], are shown. The formula and material-ids (which are allocated to the compounds in the Materials Project database [15, 26]) of the template structure that were used as a template for structure prediction with element substitution are also shown. Fig. 4.7 is same as the first 12 examples of this figure. Because the DFT calculation environment of the Materials Project and the environment described in step 3 (Section 4.2) are not fully identical, the true structures shown in this figure are the ones that were locally optimized using the DFT calculations with the environment described in step 3.

4.4 Analysis procedure for model comparison

For comparison of the metric learning models, we randomly selected 3,000 compounds for training, 1,000 compounds for validation, and 3,000 compounds for testing from the 33,153 compounds (all the stable structures in the Materials Project [15, 26], as on 11/21/2020). Then, similar to the method described in the Experimental Procedure section, 76,104, 8,234, and 80,550 pairs were obtained as training data, validation data, and test data (the ratio of similar pairs to dissimilar pairs is 1:1). Neural network binary classifier (NN-binary), neural network binary classifier (NN-regression), and Siamese network were trained five times independently, and the mean AUC and standard deviation of the AUC were calculated. The KISSME model was trained only once, as this method is not affected by random numbers. The result is shown in Fig. 4.2.

4.5 Detail of the models

The structure of the neural network models for NN-binary is shown in Table 4.6. The structure of the neural network models for NN-regression and Siamese network (precisely, each of the paired networks with shared weights) is the same as the one shown in Table 4.6, except for the top layer. The soft-max function with loss of categorical cross-entropy, the linear combination with loss of MAE, and the linear combination with contrastive loss [115] are used for NN-binary, NN-regression, and Siamese networks, respectively. In all cases, Adam [122] was used as the optimizer. In each model training, the parameters that minimized the loss in the validation data were adopted as the final parameters of the trained model. The models were trained on the training data with a batch size of 128 and number of epochs = 100.

Table 4.6. The structure of the neural network models for NN-binary. The 3 layers (unit number =50) of the fully connected network with dropout (rate = 0.2) and the rectified linear unit as activation function.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 290)	0
dense_1 (Dense)	(None, 50)	14550
dropout_1 (Dropout)	(None, 50)	0
dense_2 (Dense)	(None, 50)	2550
dropout_2 (Dropout)	(None, 50)	0
dense_3 (Dense)	(None, 50)	2550
dense_4 (Dense)	(None, 2)	102

Total params: 19,752

The structure, hyperparameters, and training procedure of the neural network models for NN- binary are identical to those used for model comparison and those used as final models in the CSP method.

4.6 Concluding remarks

We proposed a CSP method based on metric learning of crystal structure similarity. The prediction is made by selecting crystal structures that are predicted to be similar to the stable structure of a given query composition from the existing crystal structures in the database. In materials science, most crystals have been discovered by element substitution of previously discovered crystals. The proposed method can be considered as a machine learning alternative to traditional protocols in the discovery of new materials. Compared to existing methods, the most significant difference is that the proposed method does not involve any first-principles calculations, except in the final step of locally optimizing the proposed structure. Thus, the computational cost of the proposed method is significantly lower than existing methods.

Finally, we summarize the extensibility and limitations of the proposed method. Although we have focused on the prediction of stable structures, the current method may also be applicable to the prediction of metastable structures. In principle, the present method can be used to predict the identity of metastable structures in the same framework if the training instances including the metastable structures are created. The method relies on element substitution; therefore, it cannot be applied unless there is a template available for substitution. For example, as reported here, the crystal structures of approximately 3.2% (1,051/33,153) in the Materials Project have no template with the same composition ratio. Nevertheless, the present method is highly capable of identifying the template closest to the true structure present in a crystal structure database. Furthermore, as discussed in Section 4.3, at least 50–60% of all crystal systems, including unique crystals without template structures, can be predicted using the substitution-based CSP. If the crystal structure database expands monotonically in the future, the application range of the substitution-based CSP method will also expand.

5 Conclusion

In 2011, the Materials Genome Initiative (MGI) launched by the United States with the goal of halving the time required for materials development from the approximately 10–20 years taken from the discovery of a new material to its commercialization. . In its white paper, it has been declared that the development of digitalized data infrastructures and the utilization of data science techniques will be key to achieving this goal. This has brought the interdisciplinary field of materials informatics into the limelight. In Japan, the Material Research by Information Integration Initiative (MI^2I) was launched in 2015 through the support by the Japan Science and Technology Agency (JST). With the launch of this project, the National Institute for Materials Science (NIMS) has become a hub for the development of an academic infrastructure for materials informatics, the creation of human resources, and commercialization in Japan. In 2017, the Data Science Center for Creative Design and Manufacturing was established at the Institute of Statistical Mathematics (ISM). Researchers at the ISM have been working to find unique approaches to scientific problems in materials science from the unique perspective of data science, and have created and implemented new scientific methods. The present thesis is in line with this research trend.

This doctoral thesis described two of the author's scientific contributions to materials informatics, the emerging new interdisciplinary field of materials science and data science. We considered forward and inverse problems as the basic workflow of materials informatics. The objective of the forward problem is to predict the output of a system with respect to its input. On the contrary, the inverse problem predicts the input variable with the desired output by finding the inverse map of the forward prediction model. In addition, this thesis discusses the form of forward and inverse problems along the concept of representation, learning and generation of materials. The input and output variables of interest in materials science are very diverse. Because of this diversity, it is necessary to establish a data science methodology for each problem. Input variables such as chemical composition, molecule, crystal structure, etc. are numerically “represented” by descriptors, and a mathematical mapping from the input to the output is “learned” using given data. The inverse mapping of the model is then explored to “generate” materials with the desired properties in order to identify promising hypothetical materials. In this thesis, we focused on inorganic materials and address two problems related to the tasks of representation and learning.

Herein, we have tackled two fundamental problems in physical chemistry, using machine learning as a key driver. In the two studies, we discovered new problem settings in materials science from a unique perspective of data science. The task of periodic table design was formulated as a statistical visualization problem. Furthermore, the task of crystal structure prediction was formulated in the form of a metric learning problem.

The Russian chemist Mendeleev noticed that the properties of the 50 or so elements found at the time exhibited a certain periodic behavior, and he summarized the patterns in tabular form, thus inventing the prototype of the current periodic table. Interpreting the process leading to this invention from the viewpoint of data science, it can be said that Mendeleev performed "dimensionality reduction and visualization of data" by arranging the patterns of multidimensional data of elements into grid points (tables) on two-dimensional coordinates. This research aimed to answer whether machine learning can automatically design a periodic table from element data. The problem comes down to "dimensionality reduction of tabular form" of high-dimensional data. We developed an unsupervised learning method based on generative topographic mapping to reduce the data to a tabular form, and succeeded in obtaining a representation that is almost equivalent to Mendeleev's periodic table. In addition, a three-dimensional conical spiral periodic table was constructed using the proposed method. From this periodic table, we found some interesting rules that may suggest new classification criteria of elements.

In the second study, the aim was to answer whether the crystal structure can be predicted from the chemical composition of the material. Conventional approaches to crystal structure prediction are based on first-principles calculations of many-body electron systems, and the crystal structure is predicted by solving an energy minimization problem, resulting in a large computational time. We developed the crystal structure prediction algorithm based on metric learning of crystal structure similarly; the prediction is made by selecting crystal structures that are predicted to be similar to the stable structure of a given query composition from the existing crystal structures in database. In materials science, most crystals have been discovered by element substitution of previously discovered crystals, and the present method can be regarded as a machine learning alternative to such traditional protocols. The most significant difference compared to existing methods is that the present method does not involve any first-principles calculations except for the final step of locally optimizing the proposed structure, which makes it significantly less computationally expensive than the previous methods. In conclusion, the present method is highly capable of identifying the closest template to the true structure present in a crystal structure database. Furthermore, as discussed, it is estimated that 50–60% of all crystal systems, including unique crystals without template structures, can be predicted using the substitution-based crystal structure prediction. If the crystal structure database expands monotonically in the future, the application range of the substitution-based method will also expand.

References

[1] Chikyow, T. Trends in materials informatics in research on inorganic materials, *Q. Rev.*, **20**, (2006).

[2] Noh, J., Gu, G. H., Kim, S., & Jung, Y. Machine-enabled inverse design of inorganic solid materials: promises and challenges. *Chem. Sci.,* **11**, 4821 (2020).

[3] P Gómez-Bombarelli, R., Aguilera-Iparraguirre, J., Hirzel, T. D., Duvenaud, D., Maclaurin, D., Blood-Forsythe, M. A., Chae, H.S., Einzinger, M., Ha, D.G., Wu, T., & Markopoulos, G. Design of efficient molecular organic light-emitting diodes by a high-throughput virtual screening and experimental approach. *Nat. Mater.*, **15**(10), 1120–1127 (2016).

[4] Pletnev, I. V., Ivanenkov, Y. A., Tarasov, A. V. Dimensionality reduction techniques for pharmaceutical data mining. In Pharmaceutical Data Mining (Balakin, K. V., Ed.) John Wiley & Sons, Inc.: pp 423−455 (2009).

[5] Kusaba, M., Liu, C., Koyama, Y., Terakura, K., & Yoshida, R. Recreation of the periodic table with an unsupervised machine learning algorithm. *Sci. Rep*, **11**(1), 1–10 (2021).

[6] Mendeleev, D. On the relationship of the properties of the elements to their atomic weights. *Zeitschrift für Chemie*. **12**, 405–406 (1869).

[7] Bishop, C. M., Svensén, M., & Williams, C. K. I. GTM: The generative topographic mapping. *Neural Comput.* **10**, 215–234 (1998).

[8] Hohenberg, P. & Kohn, W. Inhomogeneous electron gas. *Phys. Rev.* **136**, B864–B871 (1964).

[9] Kohn, W. & Sham, L. J. Self-consistent equations including exchange and correlation effects. *Phys. Rev.* **140**, A1133–A1138 (1965).

[10] Kulis, B. Metric Learning: A Survey, *Found. Trends Mach. Learn.*, **5**, 287–364 (2013).

[11] Yamaguchi, N. GTM with latent variable dependent length-scale and variance. IEEE International Automatic Control Conference (CACS), 532–538 (2013).

[12] Seko, A., Togo, A., Hayashi, H., Tsuda, K., Chaput, L., & Tanaka, I. Prediction of low-thermal-conductivity compounds with first-principles anharmonic lattice-dynamics calculations and Bayesian optimization. *Phy. Rev. Lett.*, **115**(20), 205901 (2015).

[13] Snoek, J., Larochelle, H., & Adams, R. P. Practical Bayesian optimization of machine learning algorithms. *Adv. Neural Inf. Proc. Syst.*, **25** (2012).

[14] Williams, C. K., & Rasmussen, C. E. Gaussian processes for regression. *Adv. Neural Inf. Proc. Syst.* **5** (1996).

[15] 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 Mater.* **1**, 011002 (2013).

[16] Carrete, J., Li, W., Mingo, N., Wang, S., & Curtarolo, S. Finding unprecedentedly low-thermal-conductivity half-Heusler semiconductors via high-throughput materials modeling. *Phys. Rev. X*, **4**(1), 011019 (2014).

[17] Pilania, G., Wang, C., Jiang, X., Rajasekaran, S., & Ramprasad, R. Accelerating materials property predictions using machine learning. *Sci. Rep.*, **3**(1), 1–6 (2013)

[18] Wu, S., Kondo, Y., Kakimoto, M. A., Yang, B., Yamada, H., Kuwajima, I., Lambard, G., Hongo, K., Xu, Y., Shiomi, J., & Schick, C.. Machine-learning-assisted discovery of polymers with high thermal conductivity using a molecular design algorithm. *Npj Comput. Mater.*, **5**(1), 1-11 (2019).

[19] Otsuka, S., Kuwajima, I., Hosoya, J., Xu, Y., & Yamazaki, M. PoLyInfo: Polymer database for polymeric materials design. In 2011 IEEE International Conference on Emerging Intelligent Data and Web Technologies, pp. 22–s29 (2011).

[20] Allen, F. H., Bellard, S., Brice, M. D., Cartwright, B. A., Doubleday, A., Higgs, H., Hummelink, T., Hummelink-Peters, B. G., Kennard, O., Motherwell, W. D. S., Rodgers, J. R., & Watson, D. G. The Cambridge Crystallographic Data Center: computer-based search, retrieval, analysis and display of information. *Acta Crystallogr., Sect. B: Struct. Sci*., **35**, 2331–2339 (1979).

[21] Bergerhoff, G., Hundt, R., Sievers, R., & Brown, I. D. The inorganic crystal structure data base. *J. Chem. Inf. Comput. Sci*., **23**, 66–69 (1983).

[22] Wood, G. H., Rodgers, J. R., & Gough, S. R. Operation of an international data center: Canadian Scientific Numeric Database Service. *J. Chem. Inf. Comput. Sci,* **33**, 31–35 (1993).

[23] Villars, P., Berndt, M., Brandenburg, K., Cenzual, K., Daams, J., Hulliger, F., Massalski, T., Okamoto, H., Osaki, K., Prince, A., Putz, H., Iwata, & S. Pauling File, Binaries Edition, 1st ed., ASM International: Materials Park, Ohio. U.S.A., (2002).

[24] Gražulis, S., Chateigner, D., Downs, R. T., Yokochi, A. F. T., Quirós, M., Lutterotti, L., Manakova, E., Butkus, J., Moeck, P., & Le Bail, A. Crystallography Open Database – an open-access collection of crystal structures. *J. Appl. Crystallogr.*, **42**, 726–729 (2009).

[25] Villars, K., P. Cenzual Pearson's Crystal Data: Crystal Structure Database for Inorganic Compounds, 1st ed., ASM International: Materials Park, Ohio. U.S.A., (2017).

[26] Materials Project website. http://materialsproject.org.

[27] Curtarolo, S., Setyawan, W., Wang, S., Xue, J., Yang, K., Taylor, R. H., Nel- son, L. J., Hart, G. L. W., Sanvito, S., Buongiorno-Nardelli, M., Mingo, N., & Levy, O. AFLOWLIB.ORG: A distributed materials properties repository from high-throughput ab initio calculations. *Comput. Mater. Sci.*, **58**, 227–235 (2012).

[28] Hachmann, J., Olivares-Amaya, R., Atahan-Evrenk, S., Amador-Bedolla, C., Sánchez- Carrera, R. S., Gold-Parker, A., Vogt, L., Brockway, A. M., & Aspuru-Guzik, A. The Harvard Clean Energy Project: Large-scale computational screening and design of organic photovoltaics on the world community grid. *J. Phys. Chem. Lett,* **2**, 2241–2251 (2011).

[29] Saal, J. E., Kirklin, S., Aykol, M., Meredig, B., & Wolverton, C. Materials design and discovery with high-throughput density functional theory: The Open Quantum Materials Database (OQMD). *J. Miner. Met. Mater. Soc.*, **65,** 1501–1509 (2013).

[30] Draxl, C., Scheffler, M. NOMAD: The FAIR Concept for Big-Data-Driven Materials Science. arXiv:1805.05039 [cond-mat.mtrl-sci], 3 (2018).

[31] Seko, A., Hayashi, H., Nakayama, K., Takahashi, A., & Tanaka, I. Representation of compounds for machine-learning prediction of physical properties. *Phys. Rev. B*, **95**(14), 144110. (2017).

[32] Liu, C., Fujita, E., Katsura, Y., Inada, Y., Ishikawa, A., Tamura, R., Kimura, K., & Yoshida, R. Machine learning to predict quasicrystals from chemical compositions. *Adv. Mater.*, **33**(36), p. 2102507. (2021).

[33] XenonPy. https://github.com/yoshida-lab/XenonPy.

[34] Hart, G. L., Mueller, T., Toher, C., & Curtarolo, S. Machine learning for alloys. *Nat. Rev. Mater*, 1–26, (2021).

[35] Schütt, K. T., Arbabzadah, F., Chmiela, S., Müller, K. R., & Tkatchenko, A. Quantum-chemical insights from deep tensor neural networks. *Nat. Commun.* **8**, 13890 (2017).

[36] Callister, W. D., Jr. & Rethwisch, D. G. Materials Science and Engineering: An Introduction 10th ed. (Wiley, 2018).

[37] Zimmermann, N. E. R. & Jain, A. Local structure order parameters and site fingerprints for quantification of coordination environment and crystal structure similarity, *RSC Adv.*, 10, 6063–6081 (2020).

[38] Bartók, A. P., Kondor, R., & Csányi, G. On representing chemical environments. *Phys. Rev. B*, **87**(18), 184115 (2013).

[39] Rupp, M., Tkatchenko, A., Müller, K. R., & Von Lilienfeld, O. A. Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.*, **108**(5), 058301 (2012).

[40] Pham, T. L., Kino, H., Terakura, K., Miyake, T., Tsuda, K., Takigawa, I., & Dam, H. C. Machine learning reveals orbital interaction in materials. *Sci. Technol. Adv. Mater.*, **18**(1), 756 (2017).

[41] Sutton, C., Ghiringhelli, L.M., Yamamoto, T., Lysogorskiy, Y., Blumenthal, L., Hammerschmidt, T., Golebiowski, J.R., Liu, X., Ziletti, A. and Scheffler, M. Crowd-sourcing materials-science challenges with the NOMAD 2018 Kaggle competition. *NPJ Comput. Mater.* **5**, 111 (2019).

[42] Xie, T, & Grossman, J. C. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties, *Phys. Rev. Lett.* **120**, 145301 (2018).

[43] Noh, J., Kim, J., Stein, H. S., Sanchez-Lengeling, B., Gregoire, J. M., Aspuru-Guzik, A., & Jung, Y. Inverse design of solid-state materials via a continuous representation. *Matter*, **1**(5), 1370-1384 (2019).

[44] Hoffmann, J., Maestrati, L., Sawada, Y., Tang, J., Sellier, J. M., & Bengio, Y. Data-driven approach to encoding and decoding 3-d crystal structures. arXiv preprint arXiv:1909.00949 (2019).

[45] Kingma, D. P., & Welling, M. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114. (2013).

[46] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Proc. Syst.*, **27** (2014).

[47] 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. *Comput. Mater. Sci.*, **68**, 314–319 (2013).

[48] Ward, L., Dunn, A., Faghaninia, A., Zimmermann, N. E., Bajaj, S., Wang, Q., Montoya, J., Chen, J., Bystrom, K., Dylla, M., & Chard, K. Matminer: An open source toolkit for materials data mining. *Comput. Mater. Sci.*, **152**, 60–69 (2018).

[49] https://hackingmaterials.lbl.gov/matminer/featurizer_summary.html

[50] Schmidt, J., Marques, M. R., Botti, S., & Marques, M. A. Recent advances and applications of machine learning in solid-state materials science. *npj Comput. Mater.*, **5**(1), 1–36 (2019).

[51] Podryabinkin, E. V., Tikhonov, E. V., Shapeev, A. V., & Oganov, A. R. Accelerating crystal structure prediction by machine-learning interatomic potentials with active learning. *Phys. Rev. B*. **99**, 064114 (2019).

[52] Lyakhov, A. O., Oganov, A. R., Stokes, H. T., & Zhu., Q. New developments in evolutionary structure prediction algorithm USPEX. *Comput. Phys. Commun.* **184**, 1172–1182 (2013).

[53] Sanchez-Lengeling, B., & Aspuru-Guzik, A. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, **361**(6400), 360–365 (2018).

[54] Pickard, C. J. & Needs, R. J. High-pressure phases of silane. *Phys. Rev. Lett.* **97**, 045504 (2006).

[55] Pickard, C. J. & Needs, R. J. Structure of phase III of solid hydrogen. *Nat. Phys.* **3**, 473–476 (2007).

[56] Pickard, C. J. & Needs, R. J. Ab initio random structure searching. *J. Phys. Condens. Matter* **23**, 053201 (2011).

[57] Kirkpatrick, S. et al. Optimization by simulated annealing. *Science* **220**, 671–680 (1983).

[58] Pannetier, J., Bassas-Alsina, J., Rodriguez-Carvajal, J., & Caignaert, V. Prediction of crystal structures from crystal chemistry rules by simulated annealing. *Nature* **346**, 343–345 (1990).

[59] Wales, D. J. & Doye, J. P. K. Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms. *J. Phys. Chem. A* **101**, 5111–5116 (1997).

[60] Goedecker, S. Minima hopping: An efficient search method for the global mini- mum of the potential energy surface of complex molecular systems. *J. Chem. Phys.* **120**, 9911–9917 (2004).

[61] Amsler, M. & Goedecker, S. Crystal structure prediction using the minima hopping method. *J. Chem. Phys.* **133**, 224104 (2010).

[62] Oganov, A. R. & Glass., C. W. Crystal structure prediction using ab initio evolutionary techniques: Principles and applications. *J. Chem. Phys.* **124**, 244704 (2006).

[63] Oganov, A. R., Lyakhov, A. O., & Valle, M. How evolutionary crystal structure prediction works and why. *Acc. Chem. Res.* **44**, 227–237 (2011).

[64] Wang, Y., Lv, J., Zhu, Li., & Ma, Y. Crystal structure prediction via particle-swarm optimization. *Phys. Rev. B* **82**, 094116 (2010).

[65] Zhang, Y., Wang, H., Wang, Y., Zhang, L., & Ma, Y. Computer-assisted inverse design of inorganic electrides. *Phys. Rev. X* **7**, 011017 (2017).

[66] Yamashita, T. et al. Crystal structure prediction accelerated by Bayesian optimization. *Phys. Rev. Mater.* **2**, 013803 (2018).

[67] Terayama, K., Yamashita, T., Oguchi, T., & Tsuda, K. Fine-grained optimization method for crystal structure prediction. *npj Comput, Mater*. **4**, 32 (2018).

[68] Jacobsen, T. L., Jørgensen, M. S., & Hammer, B. On-the-fly machine learning of atomic potential in density functional theory structure optimization. *Phys. Rev. Lett.*, **120**(2), 026102. (2018).

[69] Podryabinkin, E. V., Tikhonov, E. V., Shapeev, A. V., & Oganov, A. R. Accelerating crystal structure prediction by machine-learning interatomic potentials with active learning. *Phys. Rev. B*, **99**(6), 064114 (2019).

[70] Zhu, Q., Oganov, A. R., Glass, C. W., & Stokes, H. T. Constrained evolutionary algorithm for structure prediction of molecular crystals: methodology and applications. *Acta Crystallogr., Sect. B: Struct. Sci.*, **68**(3), 215-226 (2012).

[71] Bushlanov, P. V., Blatov, V. A., & Oganov, A. R. Topology-based crystal structure generator. *Comput. Phys. Commun*., 236, 1-7 (2019).

[72] Hautier, G., Fischer, C., Ehrlacher, V., Jain, A., & Ceder, G. Data mined ionic substitutions for the discovery of new compounds. *Inorg. Chem.*, **50**(2), 656–663 (2011).

[73] Wang, H. C., Botti, S., & Marques, M. A. Predicting stable crystalline compounds using chemical similarity. *npj Comput. Mater.,* **7**(1), 1–9 (2021).

[74] Schuffenhauer, A., Ertl, P., Roggo, S., Wetzel, S., Koch, M. A., & Waldmann, H. The Scaffold Tree − Visualization of the scaffold universe by hierarchical scaffold classification. *J. Chem. Inf. Model*., **47**, 47−58 (2007).

[75] Varin, T., Schuffenhauer, A., Ertl, P., & Renner, S. Mining for bioactive scaffolds with scaffold networks: Improved compound set enrichment from primary screening data. *J. Chem. Inf. Model.*, **51**, 1528−1538 (2011).

[76] Schölkopf, B., Smola, A., & Müller, K. R. Kernel principal component analysis. International Conference on Artificial Neural Networks, 583–588 (1997).

[77] Tenenbaum, J. B., Silva, V., & Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *Science* **290**, 2319–2323 (2000).

[78] Roweis, S. T. & Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. *Science* **290**, 2323–2326 (2000).

[79] Maaten, L. & Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).

[80] Ceder, G., Morgan, D., Fischer, C., Tibbetts, K., & Curtarolo, S. Data-mining driven quantum mechanics for the prediction of structure. *MRS Bull.* **31**, 981–985 (2006).

[81] Suh, C. & Rajan, K. Data mining and informatics for crystal chemistry: establishing measurement techniques for mapping structure-property relationships. *Mater. Sci. Technol.* **25**, 466–471 (2009).

[82] Zhong, M. et al. Accelerated discovery of CO2 electrocatalysts using active machine learning. *Nature* **581**, 178–183 (2020).

[83] Gaspar, HA., Baskin, II., & Marcou, G., et al. Chemical data visualization and analysis with incremental generative topographic mapping: Big Data Challenge. *J. Chem. Inf. Model.* **55**, 84–94 (2015).

[84] Kruskal, J. B. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, **29**, 1−27 (1964).

[85] Singh, N., Guha, R., Giulianotti, M. A., Pinilla, C., Houghten, R. A., & Medina-Franco, J. L. Chemoinformatic analysis of combinatorial libraries, drugs, natural products, and molecular libraries small molecule repository. *J. Chem. Inf. Model*., **49**, 1010−1024 (2009).

[86] LeGuilloux, V., Colliandre, L., Bourg, S., Gueńegou, G., Dubois-Chevalier, J., Morin-Allory, L. Visual characterization and diversity quantification of chemical libraries: 1. Creation of delimited reference chemical subspaces. *J. Chem. Inf. Model.,* **51**, 1762−1774 (2011).

[87] Ruddigkeit, L., Awale, M., & Reymond, J.-L. Expanding the fragrance chemical space for virtual screening. *J. Cheminformatics*, **6**, 27 (2014).

[88] Kohonen, T. Self-organized formation of topologically correct feature maps. *Biol. Cybern.* **43**, 59-69 (1982).

[89] Horvath, D., Lisurek, M., Rupp, B., Kühne, R., Specker, E., von Kries, J., Rognan, D., Andersson, C. D., Almqvist, F., Elofsson, M., Enqvist, P.-A., Gustavsson, A.-L., Remez, N., Mestres, J., Marcou, G., Varnek, A., Hibert, M., Quintana, J., & Frank, R. Design of a general-purpose European compound screening library for EU-OPEN-SCREEN. *ChemMedChem*. (2014).

[90] Svensen, J. F. M. GTM: The Generative Topographic Mapping. Ph.D. Thesis, University of Aston in Birmingham, 1998.

[91] Bishop, C.M., Svenseń, M., Williams, C.K.I. Developments of the generative topographic mapping. Neurocomputing 1998, 21, 203−224.

[92] Moseley, H. G. J. The high frequency spectra of the elements. *Philos. Mag.* 1024 (1913).

[93] Bohr, N. On the constitution of atoms and molecules. *Philos. Mag.* **26**, 1 (1913).

[94] Marchese, F. T. The chemical table: an open dialog between visualization and design. 12th International Conference Information Visualisation, 75-81, https://doi.org/10.1109/IV.2008.79 (2008).

[95] The internet database of periodic tables https://www.metasynthesis.com/webbook/35_pt/pt_database.php.

[96] Scerri, E. Trouble in the periodic table. *Educat. Chem.* **49**, 13-17 (2012).

[97] Abubakr, M. An alternate graphical representation of periodic table of chemical elements. https://arxiv.org/pdf/0910.0273.pdf (2009).

[98] Katz, G. The periodic table: an eight period table for the 21st century. *Chem Educat.* **6**, 324-332 (2001).

[99] Lemes, M. R. & Pino, A. D. Periodic table of the elements in the perspective of artificial neural networks. J. Chem. Educat. 88, 1511-1514, DOI: https://doi.org/10.1021/ed100779v (2011).

[100] Zhou, Q. et al. Learning atoms for materials discovery. *Proc. Natl Acad. Sci. USA.* **115**, 6411-6417 (2018).

[101] Hasings, W. K. Monte Carlo sampling methods using Markov chains and their applications. Biometrika. 97–109 (1970).

[102] Quadrianto, N., Smola, A. J., Song, L., & Tuytelaars, T. Kernelized sorting. *IEEE Trans. Pattern Anal. Mach. Intell*., **32**(10), 1809–1821 (2009).

[103] Berkelaar, M. R package 'lpSolve'. CRAN (2015).

[104] R Development Core Team. R: A language and environment for statistical computing. http://www.R-project.org (2013).

[105] https://github.com/Minoru938/PTG.

[106] Rahm, M., Hoffmann, R., & Ashcroft, N. W. Atomic and ionic radii of elements 1-96. *Chem.: Eur. J.* **22**, 14625-14632 (2016).

[107] Breiman, L. Random forests. *Mach. Learn.* **45**, 5–32 (2001).

[108] Levina, E. & Bickel., P. J. Maximum likelihood estimation of intrinsic dimension. *Adv Neural Inf. Proc. Syst.*, **17**, 777-784 (2005)

[109] Carter, K.M., Raich, R., & Hero, A.O. On local intrinsic dimension estimation and its applications. *IEEE Trans. Sig. Proc.*, **58**(2), 650–663 (2010).

[110] Ceruti, C., Bassis, S., Rozza, A., Lombardi, G., Casiraghi, E., Campadelli, P. DANCo: Dimensionality from Angle and Norm Concentration. arXiv preprint 1206.3881 (2012).

[111] Bruske, J. & Sommer, G. Intrinsic dimensionality estimation with optimally topology preserving maps. *IEEE Trans. Pattern Anal. Mach. Intell*., **20**(5), 572–575 (1998).

[112] CSPML: https://github.com/minoru938/cspml.

[113] Musgrave, K., Belongie, S., & Lim, S. N. A metric learning reality check. In European Conference on Computer Vision (pp. 681–699). Springer, Cham. (2020).

[114] Parkhi, O. M., Vedaldi, A., & Zisserman, A. Deep face recognition. In Proceedings of the British Machine Vision Conference (BMVC), 41.1-41.12 (2015).

[115] Chopra, S., Hadsell, R., & LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Vol. 1, 539–546, (2005).

[116] Koestinger, M., Hirzer, M., Wohlhart, P., Roth, P. M., & Bischof, H. Large scale metric learning from equivalence constraints. In 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2288–2295 (2012).

[117] Kresse, G., & Furthmüller, J. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Phys. Rev. B*, **54**(16), 11169 (1996).

[118] Blöchl, P. E. Projector augmented-wave method. *Phy. Rev. B*, **50**(24), 17953 (1994).

[119] Perdew, J. P., Burke, K., & Ernzerhof, M. Generalized gradient approximation made simple. *Phys. Rev. Lett.*, **77**(18), 3865 (1996).

[120] Hanley, J. A., & McNeil, B. J. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, **143**(1), 29–36, (1982).

[121] Momma, K., & Izumi, F. VESTA 3 for three-dimensional visualization of crystal, volumetric and morphology data. *J. Appl. Crystallogr.*, **44**(6), 1272–1276 (2011).

[122] Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. https://arxiv.org/abs/1412.6980. (2014).