ARTICLE   OPEN

Distributed representations of atoms and materials for machine learning

Luis M. Antunes [1✉], Ricardo Grau-Crespo [1] and Keith T. Butler [1,2]

The use of machine learning is becoming increasingly common in computational materials science. To build effective models of the chemistry of materials, useful machine-based representations of atoms and their compounds are required. We derive distributed representations of compounds from their chemical formulas only, via pooling operations of distributed representations of atoms. These compound representations are evaluated on ten different tasks, such as the prediction of formation energy and band gap, and are found to be competitive with existing benchmarks that make use of structure, and even superior in cases where only composition is available. Finally, we introduce an approach for learning distributed representations of atoms, named SkipAtom, which makes use of the growing information in materials structure databases.

INTRODUCTION

In recent years, the study of machine learning (ML) has had a significant impact on many disciplines. Accordingly, materials science and chemistry has recently seen a surge in interest in applying the most recent advances in ML to the problems of the field[1–6]. A central problem in materials science is the rational design of materials with specific properties. Typically, useful materials have been discovered serendipitously[7]. With the advent of ubiquitous and capable computing infrastructure, materials discovery has been increasingly aided by computational chemistry, especially density functional theory (DFT) simulations[8]. Such theoretical calculations are indispensable when investigating the properties of novel materials. However, they are computationally intensive, and performing such analysis on large numbers of compounds (e.g. there are more than 10^{10} chemically sensible stoichiometric quaternary compounds possible[9]) becomes impractical with today's computing technology. Moreover, certain chemical systems, such as those with very strongly correlated electrons, or with high levels of disorder, remain a theoretical challenge to DFT[10,11].

The application of ML to materials science aims to ameliorate some of these problems, by providing alternate computational routes to properties of interest. There have been numerous examples of the successful application of ML to chemical systems. Techniques from ML have been used to predict very local and detailed properties, such as atomic and molecular orbital energies and geometries[12] or partial charges[13], and also global properties, such as the formation energy and band gap of a given compound[14–17].

For a ML algorithm to work effectively, the objects of the system of interest must be converted into faithful representations that can be consumed in a computational context. Deriving such representations has been a main focus for researchers in ML, and in the case of deep learning, such representations are typically learned automatically, as part of the training process[18]. Related to this are the concepts of unsupervised learning, where patterns in the data are derived without the use of labels or other forms of supervision[19], and semi-supervised learning, where a small amount of labeled data is combined with large amounts of

unlabeled data[20–23]. Indeed, given that most data is unlabeled, such techniques are very valuable. Some of the most successful and widely used algorithms, such as Word2Vec from the field of Natural Language Processing (NLP), use unsupervised learning to derive effective representations of the objects in the system of interest (words, in this case)[24,25].

The most basic object of interest in chemical systems is very often the atom. Thus, there have already been several investigations examining the derivation of effective machine representations of atoms in an unsupervised setting[26–28], and other investigations have aimed to learn good atomic representations in the context of a supervised learning task[29,30]. A learned representation of an atom generally takes the form of an embedding, which can be described as a relatively low-dimensional space in which higher-dimensional vectors can be expressed. Using embeddings in a ML task is advantageous, as the number of input dimensions is typically lower than if higher-dimensional sparse vectors were used. Moreover, embeddings which are semantically similar reside closer together in space, which provides a more principled structure to the input data. Such representations should allow ML models to learn a task more quickly and effectively.

A widely held hypothesis in ML is that unlabeled data can be used to learn effective representations. In this work, we introduce an approach for learning atomic representations using an unsupervised approach. This approach, which we name SkipAtom, is inspired by the Skip-gram model in NLP, and takes advantage of the large number of inorganic structures in materials databases. We also investigate forming representations of chemical compounds by pooling atomic representations. Combining vectors by various pooling operations to create representations of systems composed from parts (e.g., sentences from words) is a common technique in NLP, but apparently remains largely unexplored in materials informatics[31]. The analogy we explore here is that atoms are to compounds as words are to sentences, and our results demonstrate that effective representations of compounds can be composed from the vector representations of the constituent atoms. Finally, a common problem when searching chemical

[1]Department of Chemistry, University of Reading, Whiteknights, Reading RG6 6DX, UK. [2]SciML, Scientific Computing Department, Rutherford Appleton Laboratory, Harwell OX11 0QX, UK. ✉email: l.m.antunes@pgr.reading.ac.uk

space for new materials is that the structure of a compound may not be known. Since the properties of a material are typically tightly coupled to its structure, this creates a significant barrier[32]. Here, we compare our models, which operate on representations derived from chemical formulas only, to benchmarks that are based on models that use structural information. We find that, for certain tasks, the performance of the composition-only models is comparable.

RESULTS

Representations of atoms and compounds

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

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

Atomic representations

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

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

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

Atom2Vec. If one may know a word by the company it keeps, then the same might be said of an atom. In 2018, Zhou et al. described an approach for deriving distributed atom vectors that involves generating a co-occurrence count matrix of atoms and their chemical environments, using an existing database of materials, and applying singular value decomposition to the matrix[26]. The number of dimensions of the resulting atomic vectors is limited to the number of atoms used in the matrix.

Mat2Vec. A popular means of generating word vectors in NLP is through the application of the Word2Vec algorithm, wherein an unsupervised learning task is employed[24]. Given a corpus (a collection of text), the goal is to predict the likelihood of a word
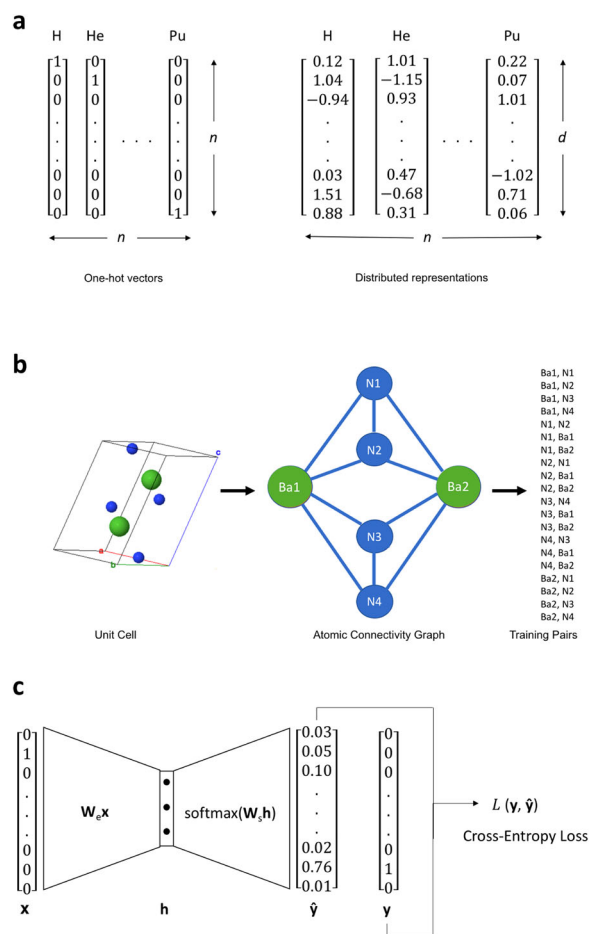

Fig. 1 Atom vectors and training process for SkipAtom. a Scheme illustrating one-hot and distributed representations of atoms. In the diagram, there are n atoms represented, and d is the adjustable number of dimensions of the distributed representation. Note that the atoms in this example are H, He and Pu, but they could be any atom. **b** Scheme describing how training data is derived for the creation of SkipAtom vectors. Here, a graph representing the atomic connectivity in the structure of Ba_2N_4 is depicted, and the resulting target-context atom pairs derived for training. The graph is derived from the unit cell of Ba_2N_4. **c** Scheme describing how the SkipAtom vectors are derived through training. Here, a one-hot vector, **x**, representing a particular atom is transformed into an intermediate vector **h** via multiplication with matrix $\mathbf{W}_e$. The matrix $\mathbf{W}_e$ is the embedding matrix, whose columns will be the final atom vectors after training. Training consists of minimizing the cross-entropy loss between the output vector $\hat{\mathbf{y}}$ and the one-hot vector representing the context atom, **y**. The output $\hat{\mathbf{y}}$ is obtained by applying the softmax function to the product $\mathbf{W}_s\mathbf{h}$.

occurring in the context of another. A neural network architecture is employed, and the learned parameters of the projection layer constitute the word vectors that result after training. In 2019, Tshitoyan et al. described an approach for deriving distributed atom vectors by making direct use of the materials science literature[27]. Instead of using a database of materials, they assembled a textual corpus from millions of scientific abstracts related to materials science research, and then applied the Word2Vec algorithm to derive the atom representations.

SkipAtom. In the NLP Skip-gram model, an occurrence of a word in a corpus is associated with the words that co-occur within a context window of a certain size. The task is to predict the context words given the target word. Although the aim is not to build a classifier, the act of tuning the parameters of the model so that it is able to predict

the context of a word results in a parameter matrix that acts effectively as the embedding table for the words in the corpus. Words that share the same contexts should share similar semantic content, and this is reflected in the resulting learned low-dimensional space. Analogously, atoms that share the same chemo-structural environments should share similar chemistry.

In the SkipAtom approach, the crystal structures of materials from a database are used in the form of a graph, representing the local atomic connectivity in the material, to derive a dataset of connected atom pairs (Fig. 1b). Then, similarly to the Skip-gram approach of the Word2Vec algorithm, Maximum Likelihood Estimation is applied to the dataset to learn a model that aims to predict a context atom given a target atom.

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

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

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

In practice, this means that the cross-entropy loss between the one-hot vector representing the context atom and the normalized probabilities produced by the model, given the one-hot vector representing the target atom, is minimized (Fig. 1c).

The graph representing a material can be derived using any approach desired, but in this work, an approach is used which is based on Voronoi decomposition[33], which identifies nearest neighbors using solid angle weights to determine the probability of various coordination environments[34,35]. (See Supplementary Note 3 for more information about how the graphs are derived.)

The result of SkipAtom training is a set of vectors, one for each atom of interest (Fig. 1a), that reflects the unique chemical nature of the represented atom, as well as its relationship to other atoms.

A complicating factor in the procedure just described is that some atoms may be under-represented in the database, relative to others. This will result in the parameters of those infrequently occurring atoms receiving fewer updates during training, resulting in lower quality representations for those atoms. This is an issue when learning word representations as well, and there have been several solutions proposed in the context of NLP[36,37]. Borrowing from these solutions, we apply an additional, optional processing step to the learned vectors, termed *induction*. The aim is to adjust the learned vectors so that they reside in a more sensible area of the representation space. To achieve this, each atom is first represented as a triple, given by its periodic table group number and row number, and its electronegativity. Then, for each atom, the closest atoms are obtained, in terms of the cosine similarity between the vectors formed from these triples. Using the learned embeddings for these closest atoms, a *mean nearest-neighbor representation* is derived, and the induced atom vector, $\hat{\mathbf{u}}$, is formed by adding the original atom vector, $\mathbf{u}$, to the mean nearest-neighbor:

$$\hat{\mathbf{u}} = \mathbf{u} + \frac{1}{N} \sum_{k=1}^{N} e^{-(k-1)} \mathbf{v}_k \tag{2}$$

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

Compound representations
Atom vectors by themselves may not be directly useful, as most problems in materials informatics involve chemical compounds.

However, atom vectors can be combined to form representations of compounds.

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

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

$$\mathbf{w} = \sum_{k=1}^{m} c_k \mathbf{v}_k \tag{3}$$

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

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

$$\mathbf{w} = \frac{\sum_{k=1}^{m} c_k \mathbf{v}_k}{\sum_{k=1}^{m} c_k} \tag{4}$$

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

$$\mathbf{w} = \max_{k=1}^{m} c_k \mathbf{v}_k \tag{5}$$

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

ElemNet (mean-pooled one-hot vectors). If we assign a unique one-hot vector to each atom, and perform mean-pooling of these vectors when forming a representation for a chemical compound, then the result is the same as the input representation for the ElemNet model[29]. Such a compound vector is sparse (as most compounds do not typically contain more than 5 or 6 atom types). Each component of the vector contains the unit normalized amount of the atom in the formula. For example, for H_2O, the component corresponding to H would have a value of 0.66 whereas the component corresponding to O would have a value of 0.33, and all other components would have a value of zero.

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

Evaluation tasks
A number of diverse materials ML tasks are utilized to evaluate the effectiveness of the pooled atom vector representations, and the quality of the SkipAtom representation. In total, ten previously

Table 1. The predictive tasks utilized in this study to evaluate the atom vector pooling approach.

Task	Type	Examples	Structure?	Method
Band Gap (eV)	Regression	4604	No	Experiment[15]
Band Gap (eV)	Regression	106,113	Yes	DFT-GGA[39,40]
Bulk Modulus (log(GPa))	Regression	10,987	Yes	DFT-GGA[41]
Shear Modulus (log(GPa))	Regression	10,987	Yes	DFT-GGA[41]
Refractive Index (n)	Regression	4764	Yes	DFPT-GGA[42]
Formation Energy (eV/atom)	Regression	275,424	Yes	DFT[29,43]
Bulk Metallic Glass Formation	Classification	5680	No	Experiment[44,45]
Metallicity	Classification	4921	No	Experiment[15]
Metallicity	Classification	106,113	Yes	DFT-GGA[39,40]

All datasets and benchmarks for the tasks above are described in[39], with the exception of the formation energy task, which is described in[29].
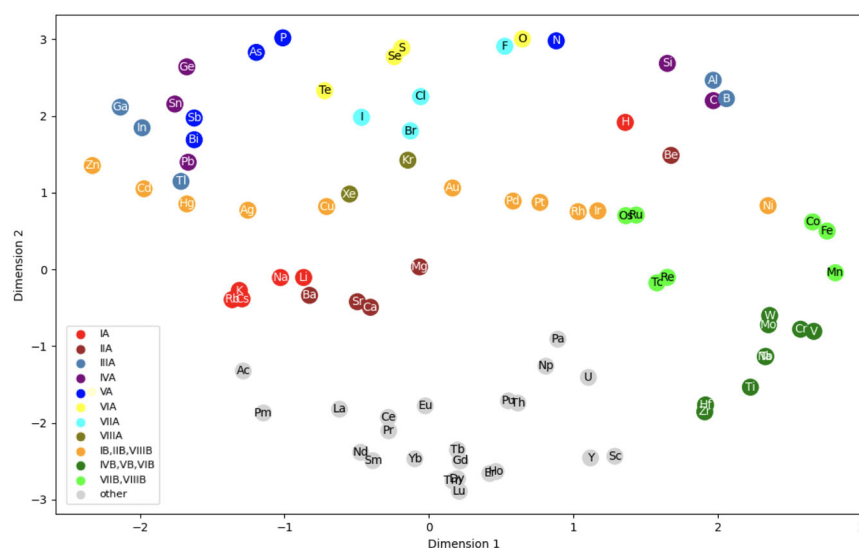

Fig. 2 Dimensionally-reduced SkipAtom vectors. Dimensionally-reduced SkipAtom atom vectors with an original size of 200 dimensions. The vectors were reduced to 2 dimensions using t-SNE. (See also Supplementary Figs. 1 and 2 for results of dimensionality reduction with PCA).

described tasks are utilized, and are broadly divided into two categories: those used for evaluating the pooling approach, and those used for evaluating the SkipAtom approach. To evaluate the pooling approach, nine tasks are chosen, and are described in Table 1.

The tasks were chosen to represent the various scenarios encountered in materials data science, such as the availability of both smaller and larger datasets, the need for either regression or classification, the availability of material structure information, and the means (experiment or theory) by which the data is obtained. The OQMD (Open Quantum Materials Database) Formation Energy task[29,38] requires a different training protocol, as it was derived from a different study than the other eight tasks that are used for the pooling approach, which were sourced from the Matbench test suite[39].

To evaluate the SkipAtom representation, the Elpasolite Formation Energy task was utilized. The task and the model were initially described in the paper that introduced Atom2Vec (an alternative approach for learning atom vectors)[26]. The task consists of predicting the formation energy of elpasolites, which are comprised of a quaternary crystal structure, and have the general formula ABC_2D_6. The target formation energies for 5645 examples were obtained by DFT[40]. The input consists of a concatenated sequence of atom vectors, each representing the A,

B, C, and D atoms. We reproduce the approach here, for comparison against the Atom2Vec results.

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

Evaluation results

A common technique for making high-dimensional data easier to visualize is t-stochastic neighbor embedding (t-SNE)[41]. Such a technique reduces the dimensionality of the data, typically to 2 dimensions, so that it can be plotted. Visualizing learned distributed representations in this way can provide some intuition regarding the quality of the embeddings and the structure of the learned space. In Fig. 2, the 200-dimensional learned SkipAtom vectors are plotted after utilizing t-SNE to reduce their dimensionality to 2. It is evident that there is a logical structure to the data. We see that the alkali metals are clustered together, as are the light non-metals, for example. The relative locations of the atoms in the plot reflect chemo-structural nuances gleaned from the dataset, and are not arbitrary.

Table 2. Elpasolite formation energy prediction results after ten-fold cross-validation; mean best formation energy MAE on the test set after 200 epochs of training in each fold.

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

Bold value represents the best result.
Batch size was 32, learning rate was 0.001. Note that Dim refers to the dimensionality of the atom vector; the size of the input vector is $4 \times$ Dim. All results were generated using the same procedure on identical train/ test folds.
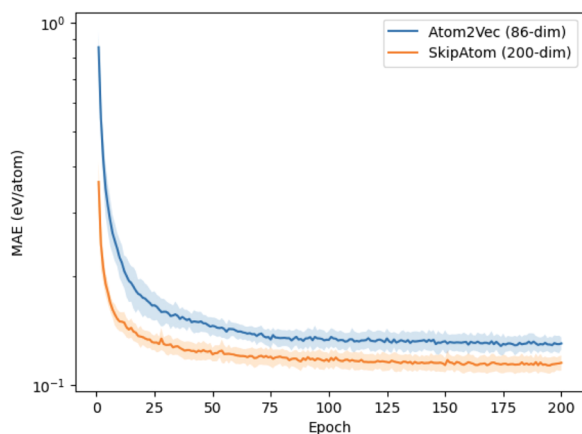

Fig. 3 Atom2Vec versus SkipAtom validation error during training. Mean absolute error during training for the Elpasolite Formation Energy prediction task, for the Atom2Vec and SkipAtom representations. The average MAE over ten folds is plotted.

To properly evaluate the quality of a learned distributed representation, they are utilized in the context of a task, and their performance compared to other representations. Here, we use the Elpasolite Formation Energy prediction task, and compare the performance of the SkipAtom vectors to the performance of other representations, namely, to Random vectors, One-hot vectors, Mat2Vec and Atom2Vec vectors. In the original study that introduced the task, atom vectors were 30- and 86-dimensional. We trained SkipAtom vectors with the same dimensions, and also with 200 dimensions, and evaluated them. The results are summarized in Table 2.

For all embedding dimension sizes, SkipAtom outperforms the other representations on the Elpasolite Formation Energy task (Mat2Vec vectors were only available for this study in 200 dimensions, and Atom2Vec vectors, by virtue of how they are created, cannot have more dimensions than atom types represented). In Fig. 3, a plot of how the mean absolute error changes during training demonstrates that the SkipAtom representation achieves better results from the beginning of training, and maintains the performance throughout.

Similar to atom vectors, compound vectors formed by the pooling of atom vectors can be dimensionally reduced, and visualized with t-SNE, or with PCA (Fig. 4a). In Fig. 4b, a sampling of several thousand compound vectors, formed by the sum-pooling of one-hot vectors, were reduced to 2 dimensions using t-SNE, and plotted. Additionally, since each compound vector represents a compound in the OQMD dataset, which contains associated formation energies, a color is assigned to each point in the plot denoting its formation energy. A clear distinction can be made across the spectrum of compounds and their formation energies. The vector representations derived from the composition of atom vectors appear to have preserved the relationship between atomic composition and formation energy.

Again, as with atom vectors, the quality of a compound vector is best established by comparing its performance in a task. To evaluate the quality of pooled atom vectors, nine predictive tasks were utilized, as described in Table 1. The performance on the benchmark regression tasks is summarized in Table 3, and the performance on the benchmark classification tasks is summarized in Table 4. Finally, the performance on the OQMD Formation Energy prediction task is summarized in Table 5.

In the benchmark regression and classification task results, there is not a clear atom vector or pooling method that dominates. The 200-dimensional representations generally appear to perform better than the smaller 86-dimensional representations. Though not evident from Tables 3 and 4, sum- and mean-pooling outperform max-pooling (see Supplementary Note 1 and Supplementary Tables 1–10). The pooled Mat2Vec representations are notable, in that they achieve the best results in 4 of the 8 benchmark tasks, while pooled SkipAtom representations are best in 2 of the 8 benchmark tasks. Pooled Random vectors tend to under-perform, though not always by a very large margin. This may not be so surprising, since random vectors exhibit quasi-orthogonality as their dimensionality increases, and thus may have the same functional characteristics as one-hot vectors[42]. On the OQMD Formation Energy prediction task, the Bag-of-Atoms representation yields the best results, significantly outperforming both the distributed representations, and the mean-pooled one-hot representation originally used in the ElemNet paper, that introduced the task.

A noteworthy aspect of these results is how the pooled atom vector representations compare to the published state-of-the-art values on the 8 benchmark tasks from the Matbench test suite. Figure 5 depicts this comparison. Indeed, the models described in this report outperform the existing benchmarks on tasks where only composition is available (namely, the Experimental Band Gap, Bulk Metallic Glass Formation, and Experimental Metallicity tasks). Also, on the Theoretical Metallicity task and the Refractive Index task, the pooled SkipAtom, Mat2Vec and one-hot vector representations perform comparably, despite making use of composition information only.

DISCUSSION

NLP researchers have learned many lessons regarding the computational representations of words and sentences. It could be fruitful for computational materials scientists to borrow techniques from the study of Computational Linguistics. Above, we have described how making an analogy between words and sentences, and atoms and compounds, allowed us to borrow both a means of learning atom representations, and a means of forming compound representations by pooling operations on atom vectors. Consequently, we draw the following conclusions: (1) effective computational descriptors of atoms can be derived from freely available and growing materials databases; (2) effective computational descriptors of compounds can be easily constructed by straightforward pooling operations of the atom vectors of the constituent atoms; (3) representations of material
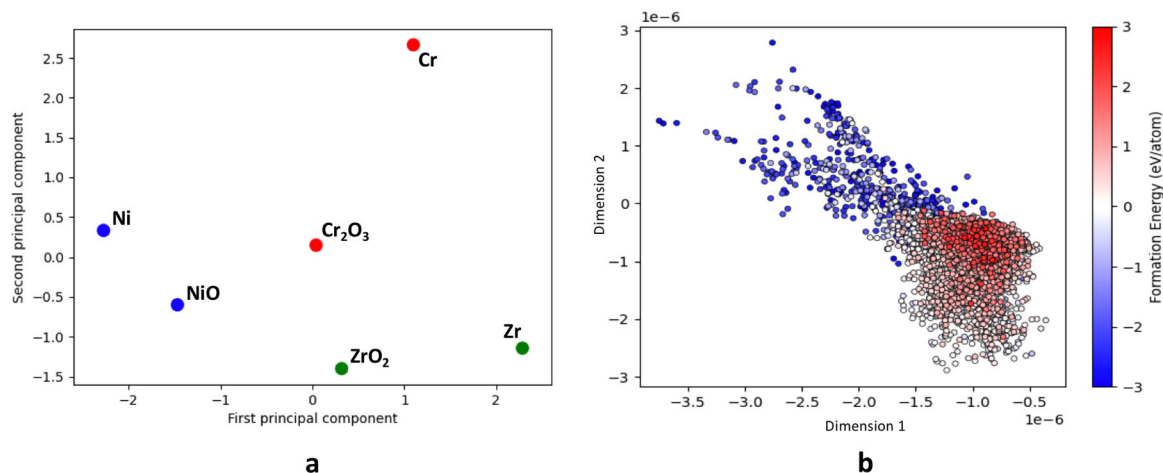
Fig. 4 Dimensionally-reduced compound vectors. a 200-dimensional SkipAtom vectors for Cr, Ni, and Zr, and their mean-pooled oxides, dimensionally reduced using PCA. **b** Plot of a sampling of the dimensionally-reduced compound vectors for the OQMD Dataset Formation Energy task, mapped to their associated physical values. The points are sum-pooled one-hot vectors reduced using t-SNE with a Hamming distance metric. The sum-pooled one-hot representation was the best performing for the task.

Table 3. Benchmark regression task results after two-repeated five- or ten-fold cross-validation; mean best MAE on the test set after 100 epochs of training in each fold.

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

All results were generated using the same procedure on identical train/test folds. TBG refers to the theoretical band gap task (MAE in eV), BM to the bulk modulus task (MAE in log(GPa)), SM to the shear modulus task (MAE in log(GPa)), and RI to the refractive index task (MAE in n). These tasks make use of structure information. EBG refers to the experimental band gap task (MAE in eV), and it makes use of composition only. Only the best results for each representation are reported. The pooling procedure varies between results; blue results represent sum-pooling, red results represent mean-pooling, and teal results represent max-pooling. Numbers in parentheses represent the standard deviation to one part in 10^4. See the Supplementary Information tables for more detailed results.

Table 4. Benchmark classification task results after two-repeated five-fold stratified cross-validation; mean best ROC-AUC on the test set after 100 epochs of training in each fold.

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

All results were generated using the same procedure on identical train/test folds. TM refers to the theoretical metallicity task, and makes use of structure information. BMGF refers to the bulk metallic glass formation task, and EM to the experimental metallicity task. These last two do not make use of structure information. Only the best results for each representation are reported. The pooling procedure varies between results; blue results represent sum-pooling, red results represent mean-pooling, and teal results represent max-pooling. See the Supplementary Information tables for more detailed results.

Table 5. OQMD dataset formation energy prediction results after ten-fold cross-validation; mean best formation energy MAE on the test set after 100 epochs of training in each fold.

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

Bold value represents the best result.
All results were generated using the same procedure on identical train/test folds.
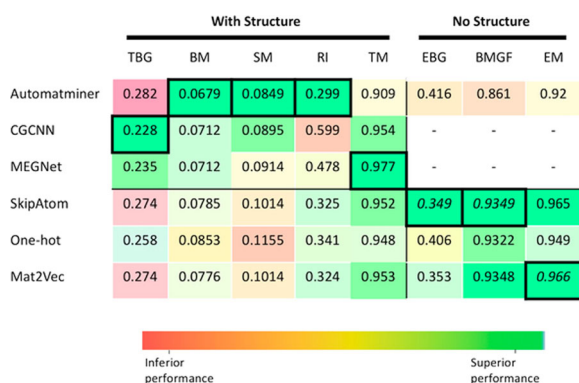
Fig. 5 Performance scores on benchmark tasks. A comparison between the results of the methods described in the current work and existing state-of-the-art results on benchmark tasks. TBG refers to the Theoretical Band Gap task (MAE in eV), BM to the Bulk Modulus task (MAE in log(GPa)), SM to the Shear Modulus task (MAE in log(GPa)), RI to the Refractive Index task (MAE in n), and TM to the Theoretical Metallicity task (ROC-AUC). These tasks make use of structure information. EBG refers to the Experimental Band Gap task (MAE in eV), BMGF to the Bulk Metallic Glass Formation task (ROC-AUC), EM to the Experimental Metallicity task (ROC-AUC). These tasks make use of composition only. The results that are outlined in bold represent the best score for that task. Italicized results represent an improvement over existing best scores. As described in the "Methods" section of this report, the same methodology was used to obtain the results for all of the algorithms in the table.

composition (without structure) can be useful for predicting certain properties, and can play a useful role in hierarchical screening studies where subsequent more expensive steps account for structure.

SkipAtom performs as well as state-of-the-art embeddings, while offering significant advantages in terms of flexibility and ease of implementation. The SkipAtom representation can be derived from a dataset of readily accessible compound structures. Moreover, the training process is lightweight enough that it can be performed on a good quality laptop on a scale of minutes to several hours (given the atom pairs). This highlights some important differences between SkipAtom and Atom2Vec and Mat2Vec. Training of the Mat2Vec representation requires the curation of millions of journal abstracts, and a subsequent classification step for retaining only the most relevant abstracts. Additionally, pre-processing of the tokens in the text must be carried out to identify valid chemical formulae through the use of

custom rules and regular expressions. On the other hand, since SkipAtom makes direct use of the information in materials databases, no special pre-processing of the chemical information is required. Although the procedures for creating Mat2Vec and SkipAtom vectors have been incorporated into publicly available software libraries, the conceptually simpler SkipAtom approach leaves little room for ambiguity that might result from manually written chemical information extraction rules. When compared to Atom2Vec, a principal difference is that SkipAtom vectors are not limited in size by the number of atom types available. This allows larger SkipAtom vectors to be trained, and, as is evident from the results described above, larger vectors generally perform better on tasks. (See Supplementary Note 5 for an analysis of embedding size.) Overall, we believe SkipAtom is a more accessible tool for computational materials scientists, allowing them to readily train expressive atom vectors on chemical databases of their choosing, and to take advantage of the growing information in these databases over time. (See Supplementary Note 6 for an analysis of training dataset size.)

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

Going forward, there are a number of different avenues that can be explored. First, the atom vectors generated using the SkipAtom approach can be explored in different contexts, such as in combination with structural information. For example, graph neural networks, such as the MEGNet architecture[43], can accept any atom representation as input. It would be interesting to see if starting with pre-trained SkipAtom vectors could improve the performance of these models, where structure information is also incorporated. (See Supplementary Note 2 for preliminary results with MEGNet.) Alternatively, chemical compound vectors formed by pooling SkipAtom vectors can be directly concatenated with vectors that contain structure information, thus complementing the pooled atom vectors with more information. A candidate for encoding structure information is the Coulomb Matrix (in vectorized form), a descriptor which encodes the electrostatic interactions between atomic nuclei[44]. Finally, one limitation of the SkipAtom approach is that it does not provide representations of atoms in different oxidation states. Since it is (often) possible to unambiguously infer the oxidation states of atoms in compounds, it is, in principle, possible to construct a SkipAtom training set of pairs of atoms in different oxidation states. The number of atom types would increase by several fold, but would still be within limits that allow for efficient training. Note that by using a motif-centric learning framework, the oxidation states of transition metal elements have been effectively learned based on local bonding environments, using a graph neural network framework[45]. It would be interesting to explore the results of forming compound representations using such vectors for atoms in various oxidation states. (See Supplementary Note 4 for a preliminary experiment demonstrating the learning of representations for Fe(II) and Fe(III).)

METHODS

Pooling approach evaluation

For the purposes of evaluation, the atom and compound vectors were utilized as inputs to feed-forward neural networks. All results for evaluating the pooling approach were obtained using a 17-layer feed-forward neural network architecture based on ElemNet[29]. The network was comprised of 4 layers with 1024 neurons, followed by 3 layers with 512 neurons, 3 layers with 256 neurons, 3 layers with 128 neurons, 2 layers with 64 neurons, and 1 layer with 32 neurons, all with ReLU activation. For regression tasks, the output layer consisted of a single neuron and linear activation. For classification tasks, the output layer consisted of a single neuron and sigmoid activation (as only binary classification was performed). Instead of using dropout layers for regularization, as in the ElemNet approach, L2 regularization was used, with a regularization constant of 10^{-5}. The goal during training was to minimize the Mean Absolute Error loss (for regression tasks), or the Binary Cross-entropy loss (for classification tasks). All pooling approach experiments utilized a mini-batch size of 32, and a learning rate of 10^{-4} along with the Adam optimizer (with an epsilon parameter of 10^{-8})[46]. As described in the paper that introduces the Matbench test set[39], k-fold cross-validation was performed to evaluate the compound vectors in regression tasks, with the same random seed to ensure the same splits were used each time. For classification tasks, stratified k-fold cross-validation was performed. As required by the benchmarking protocol, five splits were used (with the exception of the OQMD Formation Energy prediction task, which used 10 splits). Because the variance was high for some tasks after k-fold cross-validation, repeated k-fold cross-validation was performed, to reduce the variance[47]. All training was carried out for 100 epochs, and the best performing epoch was chosen as the result for that split. By following this protocol, a direct and fair comparison can be made to results reported previously using the same Matbench test set[39].

Elpasolite formation energy prediction

The results for evaluating the SkipAtom approach were obtained using the Elpasolite neural network architecture and protocol, originally described in the paper that introduces Atom2Vec[26]. The input to the neural network is a vector constructed by concatenating 4 atom vectors, representing each of the 4 atoms in an Elpasolite composition. The single hidden layer consists of 10 neurons, with ReLU activation. The output layer consists of a single neuron, with linear activation. L2 regularization was used, with a regularization constant of 10^{-5}. The goal during training was to minimize the Mean Absolute Error loss. The training protocol differs slightly in this report, and ten-fold cross-validation was performed, utilizing the result after 200 epochs of training. The same random seed was used for all experiments, to ensure the same splits were utilized. A mini-batch size of 32 was utilized, and a learning rate of 10^{-3} along with the Adam optimizer (with an epsilon parameter of 10^{-8}) was chosen[46].

SkipAtom training

Learning of the SkipAtom vectors involved the use of the Materials Project database[48]. To assemble the training set, 126,335 inorganic compound structures were downloaded from the database. Each of these structures was converted into a graph representation using an approach based on Voronoi decomposition[33–35], and a dataset of co-occurring atom pairs was derived. (See Supplementary Note 3 for more information on graph derivation.) A total of 15,360,652 atom pairs were generated, utilizing 86 distinct atom types. The architecture consisted of a single hidden layer with linear activation, whose size depended on the desired dimensionality of the learned embeddings, and an output layer with 86 neurons (one for each of the utilized atom types) with *softmax* activation. The training objective consisted of minimizing the cross-entropy loss between the predicted context atom probabilities and the one-hot vector representing the context atom, given the one-vector representing the target atom as input. Training utilized stochastic gradient descent with the Adam optimizer, with a learning rate of 10^{-2} and a mini-batch size of 1024, for ten epochs.

DATA AVAILABILITY

The data that support the findings of this study are available as follows: The materials data that was used to learn the SkipAtom embeddings are publicly available online at https://materialsproject.org/. The elpasolite formation energy training data are publicly available online at https://doi.org/10.1103/PhysRevLett.117.135502, in the Supplementary Material section. The datasets comprising the Matbench tasks are publicly available at

https://hackingmaterials.lbl.gov/automatminer/datasets.html. The Mat2Vec pre-trained embeddings are publicly available online and can be downloaded by following the instructions at https://github.com/materialsintelligence/mat2vec. The Atom2Vec embeddings are publicly available online and can be obtained from https://github.com/idocx/Atom2Vec. The processed data that is used in this study, as well as scripts for reproducing the experiments, can be found on the GitHub repository at the address https://github.com/lantunes/skipatom. Any other relevant data from this work is available from the authors upon reasonable request.

CODE AVAILABILITY

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

REFERENCES

1. Himanen, L., Geurts, A., Foster, A. S. & Rinke, P. Data-driven materials science: status, challenges, and perspectives. *Adv. Sci.* **6**, 1900808 (2019).
2. Schleder, G. R., Padilha, A. C., Acosta, C. M., Costa, M. & Fazzio, A. From DFT to machine learning: recent approaches to materials science—a review. *J. Phys. Mater.* **2**, 032001 (2019).
3. Goldsmith, B. R., Esterhuizen, J., Liu, J.-X., Bartel, C. J. & Sutton, C. Machine learning for heterogeneous catalyst design and discovery. *AIChE J.* **64**, 2311–2323 (2018).
4. Butler, K. T., Davies, D. W., Cartwright, H., Isayev, O. & Walsh, A. Machine learning for molecular and materials science. *Nature* **559**, 547–555 (2018).
5. Ramprasad, R., Batra, R., Pilania, G., Mannodi-Kanakkithodi, A. & Kim, C. Machine learning in materials informatics: recent applications and prospects. *npj Comput. Mater.* **3**, 1–13 (2017).
6. 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**, 1–36 (2019).
7. DiSalvo, F. J. Challenges and opportunities in solid-state chemistry. *Pure Appl. Chem.* **72**, 1799–1807 (2000).
8. Zunger, A. Inverse design in search of materials with target functionalities. *Nature Rev. Chem.* **2**, 1–16 (2018).
9. Davies, D. W. et al. Computational screening of all stoichiometric inorganic materials. *Chem* **1**, 617–627 (2016).
10. Duan, C., Liu, F., Nandy, A. & Kulik, H. J. Putting density functional theory to the test in machine-learning-accelerated materials discovery. *J. Phys. Chem. Lett.* **12**, 4628–4637 (2021).
11. Midgley, S. D., Hamad, S., Butler, K. T. & Grau-Crespo, R. Bandgap engineering in the configurational space of solid solutions via machine learning:(mg, zn) o case study. *J. Phys. Chem. Lett.* **12**, 5163–5168 (2021).
12. Qiao, Z., Welborn, M., Anandkumar, A., Manby, F. R. & Miller III, T. F. Orbnet: deep learning for quantum chemistry using symmetry-adapted atomic-orbital features. *J. Chem. Phys.* **153**, 124111 (2020).
13. Raza, A., Sturluson, A., Simon, C. M. & Fern, X. Message passing neural networks for partial charge assignment to metal—organic frameworks. *J. Phys. Chem. C* **124**, 19070–19082 (2020).
14. Faber, F., Lindmaa, A., von Lilienfeld, O. A. & Armiento, R. Crystal structure representations for machine learning models of formation energies. *Int. J. Quantum Chem.* **115**, 1094–1101 (2015).
15. Zhuo, Y., Mansouri Tehrani, A. & Brgoch, J. Predicting the band gaps of inorganic solids by machine learning. *J. Phys. Chem. Lett.* **9**, 1668–1673 (2018).
16. Davies, D. W., Butler, K. T. & Walsh, A. Data-driven discovery of photoactive quaternary oxides using first-principles machine learning. *Chem. Mater.* **31**, 7221–7230 (2019).
17. Artrith, N. Machine learning for the modeling of interfaces in energy storage and conversion materials. *J. Phys. Energy* **1**, 032002 (2019).
18. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
19. Hinton, G. E. et al. *Unsupervised Learning: Foundations of Neural Computation* (MIT Press, 1999).
20. Chapelle, O., Schlkopf, B. & Zien, A. *Semi-Supervised Learning* 1st edn (The MIT Press, 2010).
21. Duan, C., Liu, F., Nandy, A. & Kulik, H. J. Semi-supervised machine learning enables the robust detection of multireference character at low cost. *J Phys. Chem. Lett.* **11**, 6640–6648 (2020).

22. Zhang, Y. & Lee, A. A. Bayesian semi-supervised learning for uncertainty-calibrated prediction of molecular properties and active learning. *Chem. Sci.* **10**, 8154–8163 (2019).

23. Huo, H. et al. Semi-supervised machine-learning classification of materials synthesis procedures. *npj Comput. Mater.* **5**, 1–7 (2019).

24. Mikolov, T., Chen, K., Corrado, G. & Dean, J. Efficient estimation of word representations in vector space. Preprint at https://arxiv.org/abs/1301.3781 (2013).

25. Le, Q. & Mikolov, T. Distributed representations of sentences and documents. In *Proc. 31st International Conference on Machine Learning*, 1188–1196 (PMLR, 2014).

26. Zhou, Q. et al. Learning atoms for materials discovery. *Proc. Natl. Acad. Sci. USA* **115**, E6411–E6417 (2018).

27. Tshitoyan, V. et al. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature* **571**, 95–98 (2019).

28. Chakravarti, S. K. Distributed representation of chemical fragments. *ACS Omega* **3**, 2825–2836 (2018).

29. Jha, D. et al. ElemNet: deep learning the chemistry of materials from only elemental composition. *Sci. Rep.* **8**, 1–13 (2018).

30. Goodall, R. E. & Lee, A. A. Predicting materials properties without crystal structure: deep representation learning from stoichiometry. *Nat. Commun.* **11**, 1–9 (2020).

31. Mitchell, J. & Lapata, M. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, 236–244 (2008).

32. Meredig, B. et al. Combinatorial screening for new materials in unconstrained composition space with machine learning. *Phys. Rev. B* **89**, 094104 (2014).

33. Voronoi, G. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Premier mémoire. Sur quelques propriétés des formes quadratiques positives parfaites. *J. Reine Angew. Math.* **1908**, 97–102 (1908).

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

35. Pan, H. et al. Benchmarking coordination number prediction algorithms on inorganic crystal structures. *Inorg. Chem.* **60**, 1590–1603 (2021).

36. Pilehvar, M. T. & Collier, N. De-Conflated Semantic Representations. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1680–1690 (2016).

37. Pilehvar, M. T. & Collier, N. Inducing embeddings for rare and unseen words by leveraging lexical resources. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Vol. 2, Short Papers*, 388–393 (Association for Computational Linguistics, 2017).

38. 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). *JOM* **65**, 1501–1509 (2013).

39. Dunn, A., Wang, Q., Ganose, A., Dopp, D. & Jain, A. Benchmarking materials property prediction methods: the matbench test set and automatminer reference algorithm. *npj Comput. Mater.* **6**, 1–10 (2020).

40. Faber, F. A., Lindmaa, A., Von Lilienfeld, O. A. & Armiento, R. Machine learning energies of 2 million elpasolite (ABC2D6) crystals. *Phys. Rev. Lett.* **117**, 135502 (2016).

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

42. Kainen, P. C. & Kurkova, V. Quasiorthogonal dimension. In (Kosheleva, O., Shary, S. P., Xiang, G. & Zapatrin, R. eds) *Beyond Traditional Probabilistic Data Processing Techniques: Interval, Fuzzy etc. Methods and Their Applications*, 615–629 (Springer, 2020).

43. Chen, C., Ye, W., Zuo, Y., Zheng, C. & Ong, S. P. Graph networks as a universal machine learning framework for molecules and crystals. *Chem. Mater.* **31**, 3564–3572 (2019).

44. 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**, 058301 (2012).

45. Banjade, H. R. et al. Structure motif–centric learning framework for inorganic crystalline systems. *Sci. Adv.* **7**, eabf1754 (2021).

46. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. Preprint at https://arxiv.org/abs/1412.6980 (2014).

47. Moss, H., Leslie, D. & Rayson, P. Using J-K-fold cross validation to reduce variance when tuning NLP models. In *Proceedings of the 27th International Conference on Computational Linguistics*, 2978–2989 (Association for Computational Linguistics, 2018).

48. Jain, A. et al. The materials project: a materials genome approach to accelerating materials innovation. *APL Mater.* **1**, 011002 (2013).

AUTHOR CONTRIBUTIONS

L.M.A. conceived the project, designed and performed the experiments, and drafted the manuscript. R.G.-C. and K.T.B. supervised and guided the project. All authors reviewed, edited and approved the manuscript.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.