Learning atoms for materials discovery

Quan Zhou[a], Peizhe Tang[a], Shenxiu Liu[a], Jinbo Pan[b], Qimin Yan[b], and Shou-Cheng Zhang[a,c,1]

[a]Department of Physics, Stanford University, Stanford, CA 94305-4045; [b]Department of Physics, Temple University, Philadelphia, PA 19122; and [c]Stanford Institute for Materials and Energy Sciences, SLAC National Accelerator Laboratory, Menlo Park, CA 94025

Contributed by Shou-Cheng Zhang, June 4, 2018 (sent for review February 2, 2018; reviewed by Xi Dai and Stuart P. Parkin)

Exciting advances have been made in artificial intelligence (AI) during recent decades. Among them, applications of machine learning (ML) and deep learning techniques brought human-competitive performances in various tasks of fields, including image recognition, speech recognition, and natural language understanding. Even in Go, the ancient game of profound complexity, the AI player has already beat human world champions convincingly with and without learning from the human. In this work, we show that our unsupervised machines (Atom2Vec) can learn the basic properties of atoms by themselves from the extensive database of known compounds and materials. These learned properties are represented in terms of high-dimensional vectors, and clustering of atoms in vector space classifies them into meaningful groups consistent with human knowledge. We use the atom vectors as basic input units for neural networks and other ML models designed and trained to predict materials properties, which demonstrate significant accuracy.

atomism | machine learning | materials discovery

The past 20 y witnessed the accumulation of an unprecedentedly massive amount of data in materials science via both experimental explorations and numerical simulations (1–5). The huge datasets not only enable but also call for data-based statistical approaches. As a result, a new paradigm has emerged which aims to harness artificial intelligence (AI) and machine-learning (ML) techniques (6–10) to assist materials research and discovery. Several initial attempts have been made along this path (11–16). Most of them learned maps from materials information (input) to materials properties (output) based on known materials samples. The input or feature of materials involves descriptors of constituents: Certain physical or chemical attributes of atoms are taken, depending on the materials property under prediction (11, 14, 17). Despite the success so far, these works heavily rely on researchers' wise selection of relevant descriptors; thus the degree of intelligence is still very limited from a theoretical perspective. And practically, extra computations are usually unavoidable for machines to interpret such atom descriptors which are in the form of abstract human knowledge.

To create a higher level of AI and to overcome the practical limitation, we propose Atom2Vec in this paper, which lets machines learn their own knowledge about atoms from data. Atom2Vec considers only existence of compounds in a materials database, without reference to any specific property of materials. This massive dataset is leveraged for a learning feature in materials science in an unsupervised manner (18–23). Because of the absence of materials property labels, Atom2Vec is naturally prevented from being biased to one certain aspect. As a result, the learned knowledge can yield complete and universal descriptions of atoms in principle, as long as the dataset is sufficiently large and representative. Atom2Vec follows the core idea that properties of an atom can be inferred from the environments it lives in, which is similar to the distributional hypothesis in linguistics (24). In a compound, each atom can be selected as a target type, while the environment refers to all remaining atoms together with their positions relative to the target atom. Intuitively, similar atoms tend to appear in similar environments, which allows our Atom2Vec to extract knowledge from the associations between atoms and environments and then represent it in a vector form as discussed in the following.

Atom2Vec Workflow

We begin to illustrate the full workflow of Atom2Vec as shown in Fig. 1. To capture relations between atoms and environments, atom–environment pairs are generated for each compound in a materials dataset as the first step. Before pair generation, a more explicit definition of environment is needed, whereas atoms are represented by chemical symbols conveniently. Although a complete environment should involve both chemical composition and crystal structure as mentioned before, we take into account only the former here as a proof of concept. Under this simplification, environment covers two aspects: the count of the target atom in the compound and the counts of different atoms in the remains. As an example, let us consider the compound Bi_2Se_3 from the mini-dataset of only seven samples given in Fig. 1. Two atom–environment pairs are generated from Bi_2Se_3: For atom Bi, the environment is represented as "$(2)Se3$"; for atom Se, the environment is represented as "$(3)Bi2$." Specifically, for the first pair, "(2)" in the environment $(2)Se3$ means there are two target atoms (Bi here for the compound), while "$Se3$" indicates that three Se atoms exist in the environment. Following the notation, we collect all atom–environment pairs from the dataset and then record them in an atom–environment matrix $\boldsymbol{X}$, where its entry X_{ij} gives the count of pairs with the ith atom and the jth environment. Such a matrix for the mini-dataset is also given in Fig. 1 for illustration purposes. Clearly, each row vector gives counts with different environments for one atom, and each column vector yields counts with different atoms for one environment. According to the previously mentioned intuition, two atoms behave similarly if their corresponding row vectors are close to each other in the vector space.

Although revealing similarity to some extent, descriptions of atoms in terms of row vectors of the atom–environment matrix are still very primitive and inefficient, since the vectors can be

Significance

Motivated by the recent achievements of artificial intelligence (AI) in linguistics, we design AI to learn properties of atoms from materials data on its own. Our work realizes knowledge representation of atoms via computers and could serve as a foundational step toward materials discovery and design fully based on machine learning.

Author contributions: Q.Z., P.T., S.L., Q.Y., and S.-C.Z. designed research; Q.Z., P.T., and S.L. performed research; Q.Z., P.T., and S.L. contributed new reagents/analytic tools; Q.Z., P.T., S.L., J.P., and Q.Y. analyzed data; and Q.Z. and P.T. wrote the paper.

Reviewers: X.D., Institute of Physics, Chinese Academy of Sciences; and S.P.P., Max Planck Institute of Microstructure Physics in Halle, Martin-Luther-University Halle-Wittenberg.

The authors declare no conflict of interest.

[1]To whom correspondence should be addressed. Email: sczhang@stanford.edu.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1801181115/-/DCSupplemental.

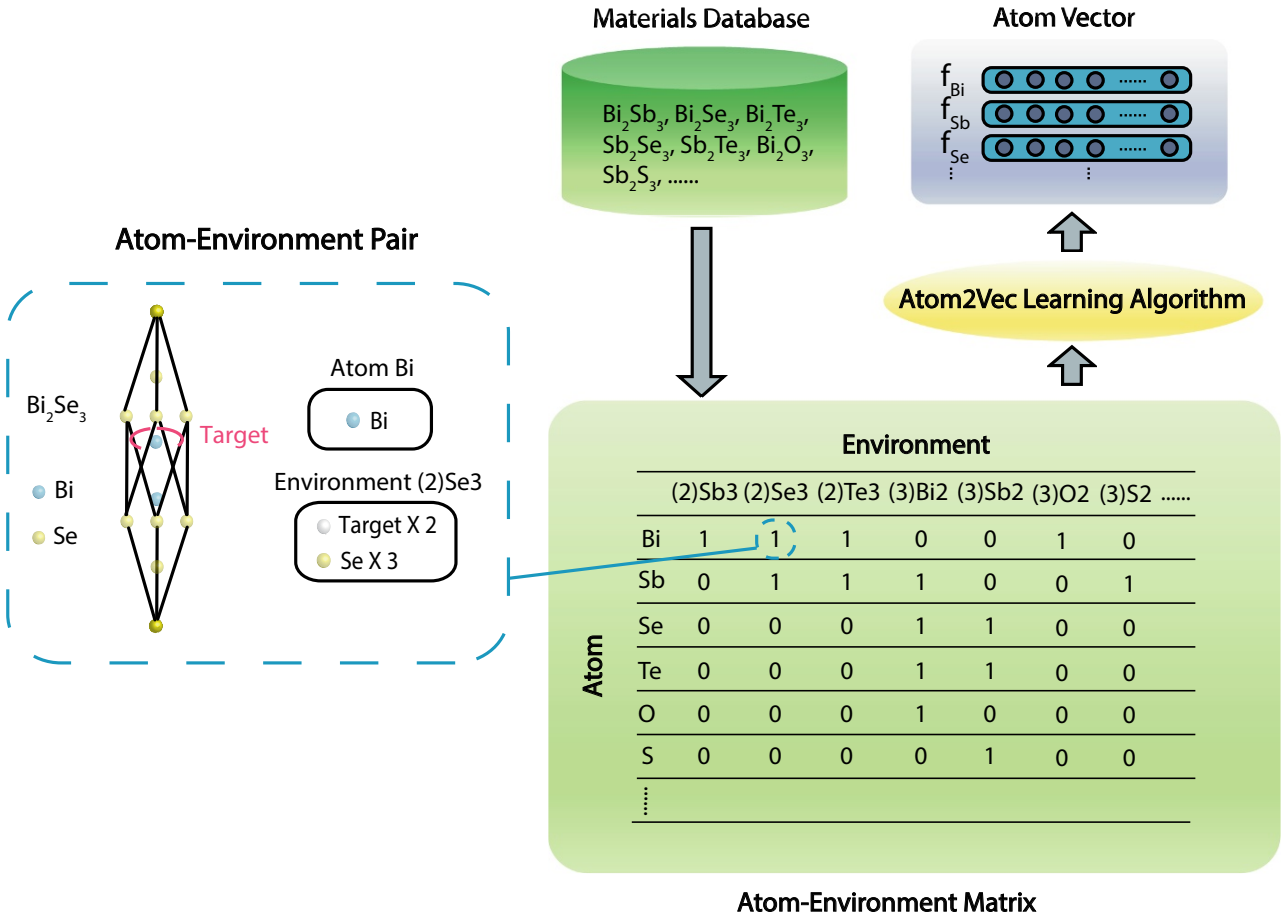

Fig. 1. Atom2Vec workflow to learn atoms from the materials database. Atom–environment pairs are generated for every compound in the materials database, based on which atom–environment matrix is constructed. A small dataset of seven compounds is used here as an example. Entries of the atom–environment matrix denote the numbers of atom–environment pairs. *Inset* shows the unit cell of compound Bi_2Se_3 and the pair corresponding to the entry of target atom *Bi* and environment (2)*Se*3. Only compositional information is considered, while structural information is ignored. Atom2Vec learning algorithms extract knowledge of atoms from the atom–environment matrix and encode learned properties in atom vectors.

extremely sparse as every atom is usually related to only a small portion of all environments. To enable machines to learn knowledge about atoms beyond the raw data statistics, we have to design algorithms that learn both atoms and environments simultaneously. The idea originates from the observation that knowing environments (atoms) could help learn atoms (environments), where underlying high-level concepts can be distilled in the collaborative process. For example, in the mini-dataset in Fig. 1, one cannot directly conclude that S and O share attributes since there is no environment shared by them (S has only environment $(3)Bi2$ while O has only environment "$(3)Sb2$" in the dataset). However, by comparing atoms associated to $(3)Bi2$ and $(3)Sb2$, one can find that the two environments are similar to each other, which in turn indicates that S and O are actually similar. Herein we have two types of learning algorithms to realize the high-level concept or knowledge extraction from the atom–environment matrix. The first type does not involve any modeling of how atom and environment interact with each other and thus is named a model-free machine. In the model-free machines, singular-value decomposition (SVD) (18) is directly applied on the reweighted and normalized atom–environment matrix (see *Materials and Methods* for details), and row vectors in the subspace of the d largest singular values encode the learned properties of atoms. The other type, the model-based machine, assumes a probability model about association between atom and environment, where randomly initialized d-dimensional vectors for atoms are optimized such that the likelihood of an existing dataset is maximized (see *Materials and Methods* for details). We proceed to analyze learned atom vectors in the next section. As our model-based machines are found to yield inferior vectors compared with model-free ones, probably due to the simplified modeling (details in *Materials and Methods*), we focus on results from model-free learning in this work, whereas atom vectors from model-based machines are involved only for comparison when necessary (details in *SI Appendix*, sections S3 and S4).

Atom Vectors. We first examine learned atom vectors for main-group elements and show that they indeed capture atoms' properties. The atom vectors learned by our model-free machine are shown in Fig. 2*A*, together with the result of a hierarchical clustering algorithm (18) based on a cosine distance metric in the vector space (see *Materials and Methods* for details). Cluster analysis is used to identify similar groups in data (18), and here we find that based on our atom vectors it manages to classify main-group elements into groups exactly the same as those in the periodic table of chemical elements (25). Active metals including alkali metals (group I) and alkali earth metals

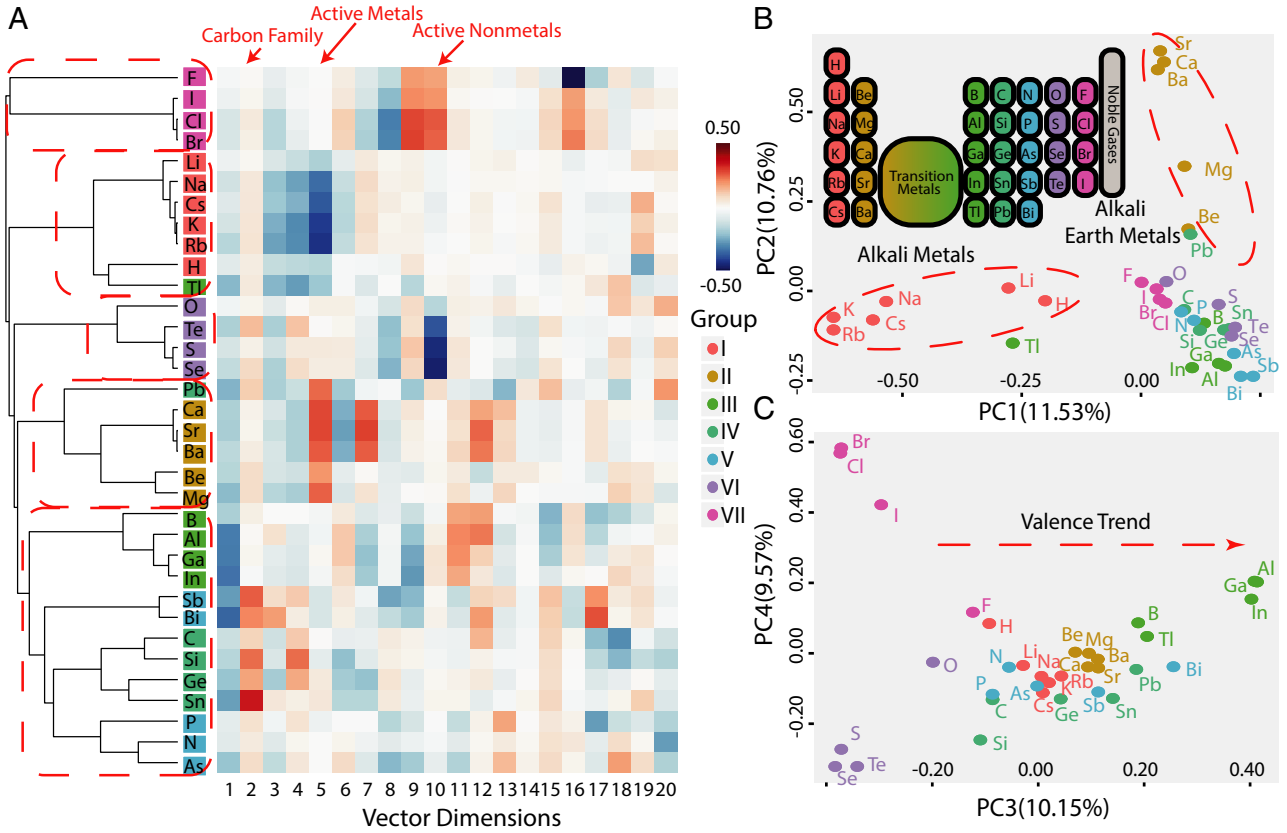

Fig. 2. Atom vectors of main-group elements learned by the model-free approach. (*A*) Illustration of atom vectors of 34 main-group elements in vector space of dimension $d = 20$ and their hierarchical clustering based on distance metric $dist(\boldsymbol{f}_1, \boldsymbol{f}_2) = 1 - \boldsymbol{f}_1 \cdot \boldsymbol{f}_2$ (*Materials and Methods*). Rows and columns denote atom types and dimension indexes, respectively; color in each cell stands for value of the vector on that dimension. Background colors of atom symbols label their columns in the periodic table. Dashed red boxes circle major atom clusters from hierarchical clustering. Red arrows point to dimensions distinguishing different types of atoms. (*B*) Projection of the atom vectors of 34 main-group elements onto the plane spanned by the first and second principal axes (PC1 and PC2). The percentage in parentheses gives the proportion of variance on that principal axis direction. *Inset* shows the periodic table of elements for reference. Dashed red circles show two distinctive clusters corresponding to two types of active metals. (*C*) Projection of atom vectors of 34 main-group elements onto the plane spanned by the third and fourth principal axes (PC3 and PC4). The percentage in parentheses gives the proportion of variance on that principal axis. Dashed red arrow indicates a valence trend almost parallel to PC3.

(group II) and active nonmetals including chalcogens (group VI) and halogens (group VII) all reside in different regions in the high-dimensional vector space. Elements in the middle of the periodic table (groups III–V) are clustered together into a larger group, indicating their similar properties. We also find in the clustering result that elements in high periods of the periodic table tend to be more metallic; for example, *Tl* from group III is close to alkali metals, and *Pb* from group IV is near alkali earth metals, both of which agree with chemical knowledge about these atoms. Moreover, there exist clear patterns in the heatmap layout of atom vectors in Fig. 2*A*, indicating that different dimensions of the vector space stand for different attributes of atoms. For instance, the 2nd dimension picks the carbon family, while the 5th dimension and the 10th dimension select active metals and active nonmetals, respectively. To better understand atom vectors in the high-dimensional vector space, we project them into several leading principal components (18) (PCs) and observe their distribution. As shown in Fig. 2 *B* and *C*, PC1 and PC2 here convincingly separate alkali metals and alkali earth metals from others. Notably, PC3 bears a moderate resemblance to valence trend among these main-group elements.

After confirming that our atom vectors learn atoms' properties, we then verify that they are more effective in use for ML materials discovery than previously widely used knowledge-based descriptors of atoms (referred to as empirical features in the following). We compare our atom vectors with those empirical features in a supervised learning task of materials prediction. The task we consider is to predict formation energies of elpasolite crystals ABC_2D_6, a type of quaternary mineral with excellent scintillation performance, which is thus very suitable for application in radiation detection (26). The dataset of this task contains nearly 10^4 elpasolites with the first-principle calculated formation energies, and the task was first introduced and modeled using kernel regression in a previous study (14). Here we use a neural network with one hidden layer to model the formation energy (Fig. 3 and *Materials and Methods*). The input layer is the concatenation of feature vectors of the four atoms (A, B, C, and D), which is then fully connected to a hidden layer equipped with nonlinear activation. The hidden layer learns to construct a representation for each compound during the supervised learning process, based on which the output formation energy is predicted. For comparison, we fix the architecture of the neural network model, feed in different types of descriptors for atoms (empirical ones and atom vectors), and examine the prediction performances (*Materials and Methods*). As in most previous studies, we use the positions of atoms in the periodic table as empirical features (14), but extend them to

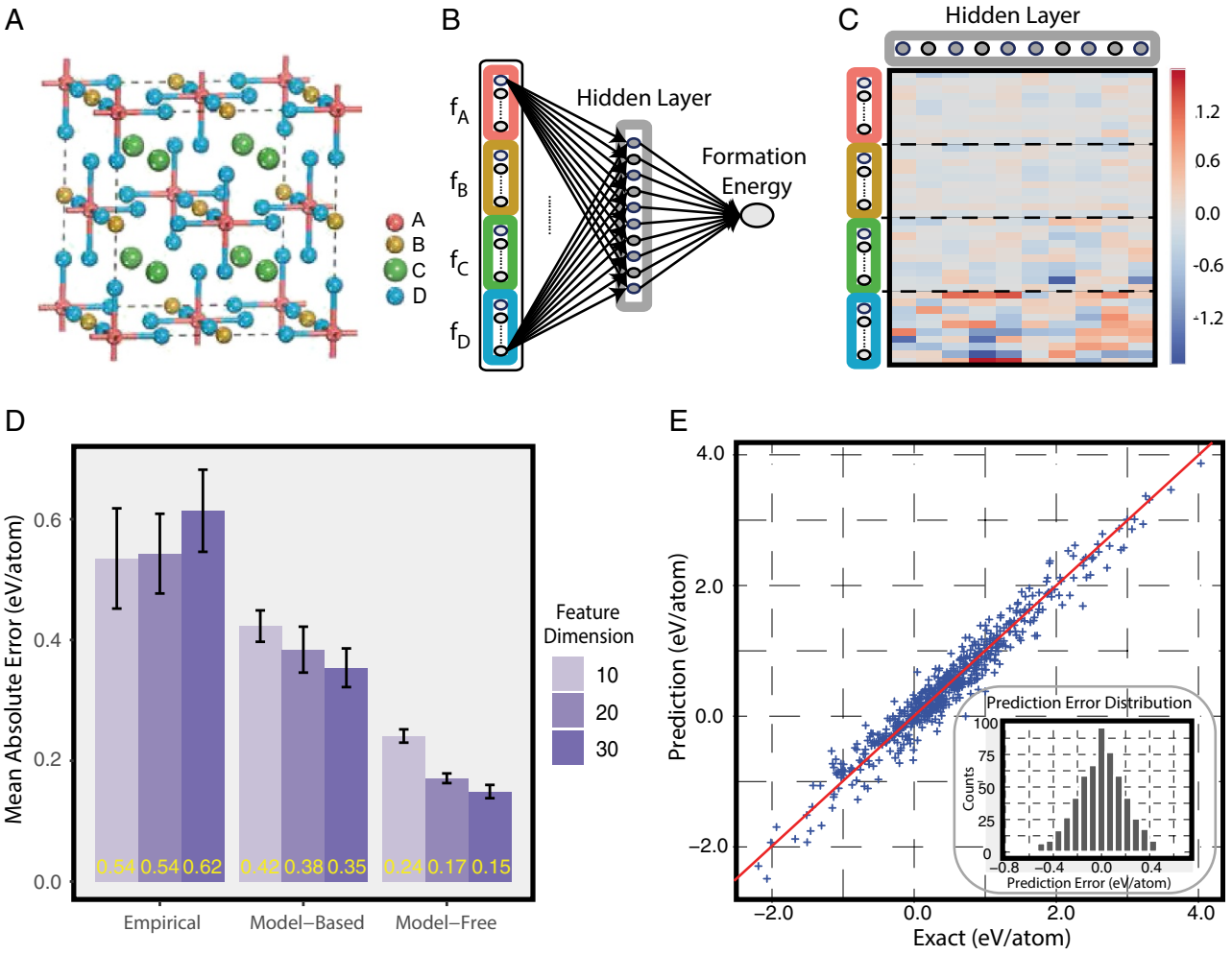

Fig. 3. Evaluation of atom vectors of main-group elements on elpasolites ABC_2D_6 formation energy prediction. (*A*) Crystal structure of elpasolites ABC_2D_6. (*B*) Architecture of the one-hidden-layer neural network for formation energy prediction. Colored boxes represent atom vectors of atoms *A*, *B*, *C*, and *D*, respectively, and gray box in the hidden layer is representation of the elpasolites compound. (*C*) Trained weights on connections between the input layer and the hidden layer in the neural network for formation energy prediction using model-free atom vectors of dimension $d = 10$. (*D*) Mean absolute test errors of formation energy prediction using different sets of atom features. Empirical features refer to the position of an atom in the periodic table, padded with random noise in expanded dimensions if necessary. Model-based features are atom vectors learned from our model-based method using an inverse-square score function (*Materials and Methods*). Model-free features are atom vectors learned from our model-free method. Error bars show the SDs of mean absolute prediction errors on five different random train/test/validation splits. (*E*) Comparison of exact formation energy and predicted formation energy using $d = 20$ model-free atom vectors. *Inset* shows the distribution of prediction errors.

the same dimension as our atom vectors by random padding for fairness. Fig. 3*D* shows the formation energy prediction errors based on empirical features, model-based learned atom vectors, and model-free learned atom vectors. The latter two clearly yield higher prediction accuracies than the empirical one, which supports the superiority of the machine-learned features over those from human knowledge. The influence of the dimension d is also examined, and it turns out that larger d leads to slight performance gain, as longer vectors could include more information. It is worth noting that, with our model-free atom vectors, the mean absolute error of formation energy prediction can be as low as 0.15 eV per atom, almost within the error range of the first-principle calculation (27). Fig. 3*E* shows the predicted formation energies vs. the exact values, and the error follows a normal distribution approximately. As a validation, we also check the neural weights between the input layer and the hidden layer in one model, which is visualized in Fig. 3*C*. The heavier weights for atom D indicate its dominant role in this problem, which agrees with previous results (14).

Our learned atom vectors not only work well for main-group elements, but also provide reasonable descriptions for other atoms including transition metals as well as functional groups. These entities are in general more versatile in properties in contrast to main-group elements, and therefore accurate description is difficult even for humans. In this part, we focus on the learned vectors of these atoms and show their advantage over empirical descriptors. These vectors along with the hierarchical clustering result are shown in Fig. 4*A*. Roughly, there appear to be three major clusters in the vector space: the transition metal cluster with Au as a representative (Au-like cluster), the cluster of lanthanoids and actinoids (lanthanoid cluster), and the transition metal cluster with Fe as a representative (Fe-like cluster). The strong amplitudes on the 14th dimension indicate Au-like and Fe-like clusters of transition metals, while the

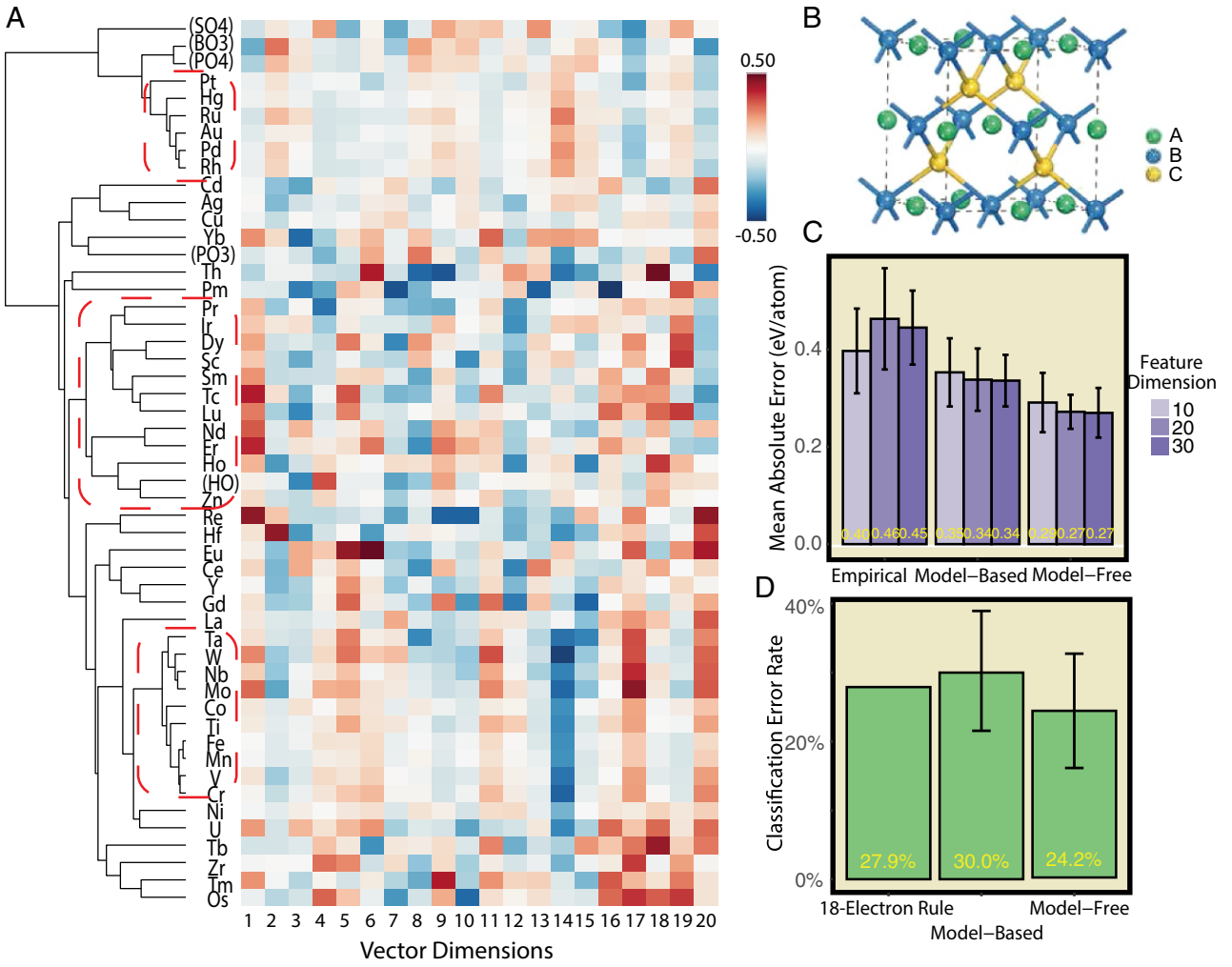

Fig. 4. Atom vectors of functional groups and elements beyond main groups learned by the model-free approach and the evaluation on tasks of half-Heusler compounds. (*A*) Illustration of atom vectors of non–main-group elements and functional groups in vector space of dimension $d = 20$ and their hierarchical clustering based on distance metric $dist(\boldsymbol{f}_1, \boldsymbol{f}_2) = 1 - \boldsymbol{f}_1 \cdot \boldsymbol{f}_2$ (*Materials and Methods*). Rows and columns denote atom types and dimension indexes, respectively; color in each cell stands for value of the vector on that dimension. Dashed red boxes circle major clusters in hierarchical clustering. (*B*) Crystal structure of half-Heusler alloys ABC. (*C*) Mean absolute test errors of formation energy prediction given by ridge regression using different sets of atom features. (*D*) Mean classification error rates of metal/insulator classifications given by the 18-electron rule and logistic regression with model-based and model-free atom vectors of dimension $d = 20$.

lanthanoid cluster shows quite a lot of hot and cold spots on other dimensions. We also note that all atoms from *Fe*-like clusters share relatively strong amplitudes on the fifth dimension, which is very similar to those of group II atoms mentioned previously. This accounts for the common +2 valence state adopted by these transition metals (25). Moreover, the atom vectors are compared with empirical ones in prediction tasks on half-Heusler alloys. These types of compounds have received a lot of research attention, because their peculiar band structures enable tunable realization of topological insulators and semimetals (28). Our tasks here include both formation energy prediction and metal/insulator classification, which are two crucial steps for topological materials search. We have nearly 250 half-Heusler compounds ABC under space group 216, along with their calculated formation energies and band gaps. Since the number of samples is limited, we use relatively simpler models here: ridge regression for prediction and logistic regression for classification (18). As shown in Fig. 4*C*, both model-free and model-based atom vectors outperform empirical ones in formation energy prediction, and the best set of vectors achieves mean absolute error 0.27 eV per atom. In metal/insulator classification as shown in Fig. 4*D*, we compare the logistic regression model using our atom vectors with the famous 18-electron rule (28). Comparable accuracies can be achieved here, which again supports the effectiveness of atom vectors in the ML tasks.

Summary and Outlook. We introduce unsupervised learning of atoms from a database of known existing materials and show the rediscovery of the periodic table by AI. The learned feature vectors not only capture well the similarities and properties of atoms in a vector space, but also show their superior effectiveness over simple empirical descriptors when used in ML problems for materials science. While empirical descriptors are usually designed specifically for a task, our learned vectors from unsupervised learning should be general enough to be applied to many cases. We anticipate their effectiveness and broad applicability can greatly boost the data-driven approaches in today's

materials science, especially for the recently proposed deep neural network methods (29–32), the same as the huge success of word vectors in language modeling (22, 23, 33, 34). Several directions related to the feature learning methods here are worthy to explore in the future. For example, the element–environment matrix can be generalized to a higher-order tensor, where the extra orders depict different parts of the composition. Such a tensor should contain finer information than the matrix, and how to extract features from this high-order object is still an open question. Also, more reasonable environment descriptions are necessary for improvement in both model-free and model-based methods. Structural information has to be taken into account to accurately model how atoms are bound together to form either environment or compound, where the recent development on recursive and graph-based neural networks (30, 32, 34) might help.

Materials and Methods

Data Preprocessing. All inorganic compounds from the Materials Project database (3) are used for our unsupervised feature learning of atoms. Compounds including more than four types of elements (or symbols in the following, to include functional groups in a more general sense) are screened out for simplicity; only binary, ternary, and quaternary compounds are selected, and they compose nearly 90% of the entire dataset of inorganic compounds (about 60,000) (*SI Appendix*, section S1). There exist compounds appearing more than once in the database, and all of them are kept since duplications give higher confidence which is meaningful in learning. No physical or chemical properties of compounds in the database are used, and as mentioned in the main text, the structural aspects are further ignored for simplicity. In other words, we build feature vectors of atoms merely from chemical formulas of all existing compounds. Some symbols are rare in terms of their number of appearances in compounds; they contribute very limited information and could impair feature learning due to high variance. So we collect counts of all symbols in compounds and consider only common ones whose counts are beyond the specified threshold (1% of the maximum count of a symbol). Several atom–environment pairs can be generated from one compound formula. Symbols of the same type are assumed to be the same in a compound, so "atom" here literally represents the type of the target atom, and "environment" includes the number of target atoms and atoms of all other elements in the compound. Therefore, the number of generated atom–environment pairs is equivalent to the number of symbol types in the compound formula.

Model-Free Methods. Our model-free methods are based on an atom–environment matrix built from atom–environment pairs. As the first step, we scan all of the pairs and count the number of pairs where the ith atom and the jth environment appear together. These counts give the entry of atom–environment matrix $X_{i,j}$, where i ranges from 1 to N while j is between 1 to M. Typically, the number of atom types N is about 100, and the number of environments M takes the order of tens of thousands, depending on the dataset ($N = 85$ and $M = 54,032$ in our case). The sum of counts over the column $\sum_j X_{i,j}$ gives the population of the ith atom, which can differ greatly among all symbols. To remove the influence of such imbalance, we focus on common symbols as mentioned above and apply the normalization $\mathcal{X}_{i,j} = X_{i,j}/(\sum_j X_{i,j}^p)^{\frac{1}{p}}$ on row vectors, where p is an integer hyperparameter that tunes the relative importance of rare environments with respect to common ones. In this way, larger p emphasizes more the role of common environments while smaller p tends to put equal attention on every environment. We test different choices of p and select $p = 2$, as it provides a natural distance metric in the vector space: The inner product of two normalized vectors, equivalently the cosine of the angle spanned by the two directions, denotes the similarity of the two symbols because it corresponds to a matching degree on all environments. The distance metric of the pair of normalized vectors $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ is explicitly defined as $dist(\boldsymbol{u}_1, \boldsymbol{u}_2) = 1 - \boldsymbol{u}_1 \cdot \boldsymbol{u}_2$.

The row vectors of the normalized matrix $\boldsymbol{\mathcal{X}} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N]^T$ provide a primitive representation for atoms. Essentially, each vector $\boldsymbol{x}_i$ gives the distribution profile over all environments, and vectors of similar atoms are close to one another in the high-dimensional space because they are likely to appear in the same environments. To have more efficient and interpretable representation for atoms, we apply SVD on the normalized matrix $\boldsymbol{\mathcal{X}}$ and project original row vectors to dimensions of leading singular values. Specifically, we factorize the $N \times M$ matrix as $\boldsymbol{\mathcal{X}} = \boldsymbol{UDV}^T$, where $\boldsymbol{U}$ is the $N \times N$ orthogonal matrix, $\boldsymbol{V}$ is the $M \times M$ orthogonal matrix, and $\boldsymbol{D}$ is the $N \times M$ diagonal matrix with diagonal elements corresponding to singular values. We select d largest singular values (*SI Appendix*, section S2) from $\boldsymbol{D}$, say $d \times d$ matrix $\tilde{\boldsymbol{D}}$, and the corresponding columns from $\boldsymbol{U}$, namely $N \times d$ matrix $\tilde{\boldsymbol{U}}$; the product of the two matrices yields an $N \times d$ matrix, whose row vectors yield better descriptions for atoms:

$$\boldsymbol{F} = \tilde{\boldsymbol{U}}\tilde{\boldsymbol{D}} = [\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_N]^T. \quad [1]$$

In contrast to the primitive representations, these feature vectors $\boldsymbol{f}_i$ are more compact and describe elements in an abstract way. As another hyperparameter, The dimension d of the feature vectors is selected by a threshold on singular values (*SI Appendix*, section S2).

Because SVD almost preserves the structure of the inner product, the distance metric mentioned previously also applies to our atom vectors. Therefore, we perform hierarchical clustering analysis of these atom vectors of dimension d based on this metric. Hierarchical clustering is used to build a hierarchy of clusters of the atoms according to a distance measure or similarity measure. Here we take a bottom–up approach in this work. In detail, every atom vector starts in its own cluster, and the distance value is initialized as zero. Then the distance value is gradually increased, and when the distance of two clusters is smaller than the value, they are merged together as a new cluster. We take a single linkage approach here; namely the minimal distance between vectors from two clusters is used as the distance between the clusters. Thus, as the distance value becomes larger, all atoms find the clusters they belong to and they are merged into one big cluster eventually. The resulting dendrogram shows the merge process and the clustering of atoms with similar properties.

Model-Based Methods. Our model-based methods rely on more assumptions about representations of environments in addition to atom–environment pairs. We represent each environment in the same vector space of atom vectors, and the composition model maps the collection of all atoms in the environment to a feature vector for the environment. Suppose feature vectors of atoms are of dimension d, which are given by the $N \times d$ matrix $\boldsymbol{F} = [\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_N]^T$. Consider one atom–environment pair whose environment includes k atoms (here k is the number of atoms except the single target atom); the atom type indexes of these atoms are $i_1, i_2, \ldots, i_k$. Based on the assumed composition model C, the environment is represented as

$$\boldsymbol{f}^{env} = C(\boldsymbol{f}_{i_1}, \boldsymbol{f}_{i_2}, \ldots, \boldsymbol{f}_{i_k}). \quad [2]$$

As an example, for the environment (2)*Se*3 from compound Bi_2Se_3, $k = 4$; namely the environment is composed of one *Bi* and three *Se*, and the environment is thus

$$\boldsymbol{f}^{env} = C(\boldsymbol{f}_{Bi}, \boldsymbol{f}_{Se}, \boldsymbol{f}_{Se}, \boldsymbol{f}_{Se}). \quad [3]$$

There are many choices for C to fully characterize compositions, for example, recursive neural networks and graph-based models (32, 34). In this work, we select the composition model C to be a summation over all atoms $\boldsymbol{f}^{env} = \sum_k \boldsymbol{f}_{i_k}$. Although this choice is oversimplified, it seems to already capture some major rules in composition (*SI Appendix*, sections S3 and S4). Thus, it is used here as a proof of concept, and more flexible and general composition models are worth testing in the future.

Another component in our model-based methods is a score function that evaluates the existence likelihood of an atom–environment pair. The score function S received an atom vector $\boldsymbol{f}_i$ and an environment vector $\boldsymbol{f}^{env}$ as built above and then returns a nonnegative value $S(\boldsymbol{f}_i, \boldsymbol{f}^{env})$, indicating the probability for the pair to exist. A larger score means that such a pair is more likely to appear. Several score functions are designed following the basic intuition that an atom and an environment should behave oppositely to be combined as a stable compound. These are the bilinear score $S(\boldsymbol{f}_i, \boldsymbol{f}^{env}) = \exp(-\boldsymbol{f}_i \cdot \boldsymbol{f}^{env})$, the Gaussian-like score $S(\boldsymbol{f}_i, \boldsymbol{f}^{env}) = \exp(-||\boldsymbol{f}_i + \boldsymbol{f}^{env}||^2)$, and the inverse-square score $S(\boldsymbol{f}_i, \boldsymbol{f}^{env}) = \frac{1}{||\boldsymbol{f}_i + \boldsymbol{f}^{env}||^2}$.

In practice, a normalized score is assigned to each atom type given an environment

$$s_i = \frac{S(\boldsymbol{f}_i, \boldsymbol{f}^{env})}{\sum_l S(\boldsymbol{f}_l, \boldsymbol{f}^{env})}. \quad [4]$$

The desired atom vectors need to maximize the average normalized score over the full dataset of atom–environment pairs or minimize the following loss function,

$$loss = \mathbb{E}_{dataset}[-\ln s(\boldsymbol{f}_{i_e}, C(\boldsymbol{f}_{i_1}, \boldsymbol{f}_{i_2}, \ldots, \boldsymbol{f}_{i_k}))], \quad [5]$$

where $\boldsymbol{f}_i$ stands for the target atom in each pair, and $\boldsymbol{f}_{i_1}, \boldsymbol{f}_{i_2}, \ldots, \boldsymbol{f}_{i_k}$ are for atoms in the environment. So our model-based feature learning is now cast into a optimization problem. To solve it, we first randomly initialize all feature vectors of atoms, and then a mini-batch stochastic gradient descent method is applied to minimize the loss function (*SI Appendix*, section S3). We choose the batch size to be 200, the learning rate to be 0.001, and the number of training epochs to be around 100. Feature vectors of different dimensions d are learned following the model-based methods.

Models in Prediction Tasks. We briefly introduce the ML methods used in our feature evaluation tasks. A neural network model is used to predict the formation energy of elpasolites ABC_2D_6. Initially inspired by neural science, in neural networks, artificial neurons are arranged layer by layer, and adjacent layers are fully connected with a linear transformation. Nonlinearity functions [such as sigmoid unit and rectified gated linear unit (ReLU)] are applied on neurons on intermediate layers. Prediction, either regression or classification, is made according to the output layer. The weights and bias between layers are optimized to minimize the loss function over the training dataset, and the loss function over a holdout dataset is used as a validation that prevents overfitting and guarantees model generalizability. We train a neural network with one hidden layer for formation energy prediction in this work. The input layer $\boldsymbol{l}$ is a concatenation of the feature vectors of the four atom types in the compound, and its dimension is $4d$, where d is the dimension of feature vectors. The hidden layer $\boldsymbol{h}$ contains 10 neurons, which produces a representation of the compound. Formation energy is given by a weighted linear combination of these hidden units. Explicitly, the model is written as

$$\boldsymbol{h} = Relu(\boldsymbol{W} \cdot \boldsymbol{l} + \boldsymbol{b}), \quad [6]$$

$$E_{formation} = \boldsymbol{w} \cdot \boldsymbol{h} + e, \quad [7]$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are weights and bias between the input layer and the hidden layer, and $\boldsymbol{w}$ and e are weights and bias between the hidden layer and the output layer. Rectified linear unit function $Relu(z) = \Theta(z)z$ gives nonlinear activation here, where $\Theta(z)$ is Heaviside step function. The mean-square error of formation energy prediction is used as a loss function. There are 5,645 ABC_2D_6 compounds with known formation energies, and we randomly hold out 10% as a test set, 80% as a training set, and the other 10% as a validation set. The training is terminated once the validation loss does not decrease anymore; the trained model is then evaluated on the holdout test set. We train and evaluate the model on 10 different random train–test splits, and the average of mean absolute errors on test sets is reported for each set of features.

For the task for half-Heusler compounds, ridge regression and logistic regression are adopted. Ridge regression is a variant of vanilla linear regression that adds an $L2$ regularization term (18) to the original mean-square error loss function. This term prevents overfitting of the training data and thus improves generalizability of the model. Logistic regression generalizes linear regression into a method for binary category classification. The sigmoid function is applied on the output unit, and a log-likelihood loss function replaces the mean-square error loss. We apply ridge regression and logistic regression to formation energy prediction and metal/insulator classification, respectively. The same data splits as above are taken, and we report the average mean absolute error and classification error rate for models using different sets of features.

ACKNOWLEDGMENTS. Q.Z., P.T. and S.-C.Z. acknowledge the Department of Energy, Office of Basic Energy Sciences, Division of Materials Sciences and Engineering, under Contract DE-AC02-76SF00515. J. P. and Q.Y. are supported by the Center for the Computational Design of Functional Layered Materials, an Energy Frontier Research Center funded by the US Department of Energy, Office of Science, Basic Energy Sciences under Award DE-SC0012575.

1. Kang B, Ceder G (2009) Battery materials for ultrafast charging and discharging. *Nature* 458:190–193.
2. Norskov JK, Bligaard T, Rossmeisl J, Christensen CH (2009) Towards the computational design of solid catalysts. *Nat Chem* 1:37–46.
3. Jain A, et al. (2013) The materials project: A materials genome approach to accelerating materials innovation. *APL Mater* 1:011002.
4. Curtarolo S, et al. (2013) The high-throughput highway to computational materials design. *Nat Mater* 12:191–201.
5. Agrawal A, Choudhary A (2016) Perspective: Materials informatics and big data: Realization of the fourth paradigm of science in materials science. *APL Mater* 4:053208.
6. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems 25*, eds Pereira F, Burges CJC, Bottou L, Weinberger KQ (Curran Associates, Inc., Dutchess County, NY), pp 1097–1105.
7. Hinton G, et al. (2012) Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Process Mag* 29:82–97.
8. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. *Proceedings of the 27th International Conference on Neural Information Processing Systems, NIPS'14*, eds Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ (MIT Press, Cambridge, MA), pp 3104–3112.
9. Silver D, et al. (2016) Mastering the game of Go with deep neural networks and tree search. *Nature* 529:484–489.
10. Silver D, et al. (2017) Mastering the game of Go without human knowledge. *Nature* 550:354–359.
11. Rupp M, Tkatchenko A, Müller KR, von Lilienfeld OA (2012) Fast and accurate modeling of molecular atomization energies with machine learning. *Phys Rev Lett* 108:058301.
12. Meredig B, et al. (2014) Combinatorial screening for new materials in unconstrained composition space with machine learning. *Phys Rev B* 89: 094104.
13. Gómez-Bombarelli R, et al. (2016) Design of efficient molecular organic light-emitting diodes by a high-throughput virtual screening and experimental approach. *Nat Mater* 15:1120–1127.
14. Faber FA, Lindmaa A, von Lilienfeld OA, Armiento R (2016) Machine learning energies of 2 million elpasolite *ABC2D6* crystals. *Phys Rev Lett* 117:135502.
15. Xue D, et al. (2016) Accelerated search for materials with targeted properties by adaptive design. *Nat Commun* 7:11241.
16. Raccuglia P, et al. (2016) Machine-learning-assisted materials discovery using failed experiments. *Nature* 533:73–76.
17. Faber F, Lindmaa A, Anatole von Lilienfeld O, Armiento R (2015) Crystal structure representations for machine learning models of formation energies. arXiv:1503.07406v1.
18. Hastie T, Tibshirani R, Friedman J (2001) *The Elements of Statistical Learning, Springer Series in Statistics* (Springer, New York).
19. Bishop CM (2006) *Pattern Recognition and Machine Learning*, Information Science and Statistics (Springer, New York).
20. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521:436–444.
21. Goodfellow I, Bengio Y, Courville A (2016) *Deep Learning* (MIT Press, Cambridge, MA).
22. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems 26*, eds Pereira F, Burges CJC, Bottou L, Weinberger KQ (Curran Associates, Inc., Dutchess County, NY), pp 3111–3119.
23. Pennington J, Socher R, Manning CD (2014) Glove: Global vectors for word representation. *Empirical Methods in Natural Language Processing (EMNLP)*. Available at https://www.aclweb.org/anthology/D14-1162. Accessed October 15, 2017.
24. Harris ZS (1954) Distributional structure. *WORD* 10:146–162.
25. Greenwood N, Earnshaw A (1997) *Chemistry of the Elements* (Butterworth-Heinemann, Oxford).
26. Hawrami R, Ariesanti E, Soundara-Pandian L, Glodo J, Shah KS (2016) *Tl2LiYCl6 : Ce*: A new elpasolite scintillator. *IEEE Trans Nucl Sci* 63:2838–2841.
27. Ghiringhelli LM, Vybiral J, Levchenko SV, Draxl C, Scheffler M (2015) Big data of materials science: Critical role of the descriptor. *Phys Rev Lett* 114:105503.
28. Chadov S, et al. (2010) Tunable multifunctional topological insulators in ternary Heusler compounds. *Nat Mater* 9:541–545.
29. Kearnes S, McCloskey K, Berndl M, Pande V, Riley P (2016) Molecular graph convolutions: Moving beyond fingerprints. *J Comput Aided Mol Des* 30:595–608.
30. Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE (2017) Neural message passing for quantum chemistry. arXiv:1704.01212v2.
31. Faber FA, et al. (2017) Fast machine learning models of electronic and energetic properties consistently reach approximation errors better than DFT accuracy. arXiv:1702.05532v2.
32. Schütt KT, Arbabzadah F, Chmiela S, Müller KR, Tkatchenko A (2017) Quantum-chemical insights from deep tensor neural networks. *Nat Commun* 8:13890.
33. Landauer TK, Dumais ST (1997) A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychol Rev* 104:211–240.
34. Socher R, et al. (2013) Recursive deep models for semantic compositionality over a sentiment treebank. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Available at http://www.aclweb.org/anthology/D13-1170. Accessed October 15, 2017.

PNAS
www.pnas.org

Supplementary Information for

Atom2Vec: learning atoms for materials discovery

Quan Zhou, Peizhe Tang, Shenxiu Liu, Jinbo Pan, Qimin Yan, and Shou-Cheng Zhang

Shou-Cheng Zhang
Email: sczhang@stanford.edu

This PDF file includes:

Supplementary text
Figs. S1 to S5
References for SI reference citations

Other supplementary materials for this manuscript include the following:

www.pnas.org/cgi/doi/10.1073/pnas.1801181115

S1. Statistics of compound data

There are 60605 inorganic compounds in total in Materials Project database[1]. Because environments with too many atom types usually depends on structures heavily, they do not help and even impair atom learning if only chemical formulas are taken into account. Hence, as mentioned in Method, only binary, ternary and quaternary compounds are used to learn atom vectors in this work. The population of each symbol (equivalently, the number of environments with this symbol as target atom) is also examined. It is found that there is a sharp population drop between common and rare symbols. A threshold, 1% of the maximal population which is located near the drop, is selected to filter out rare symbols. Only atom-environment pairs of common symbols (about 100) are used to learn atom vectors.

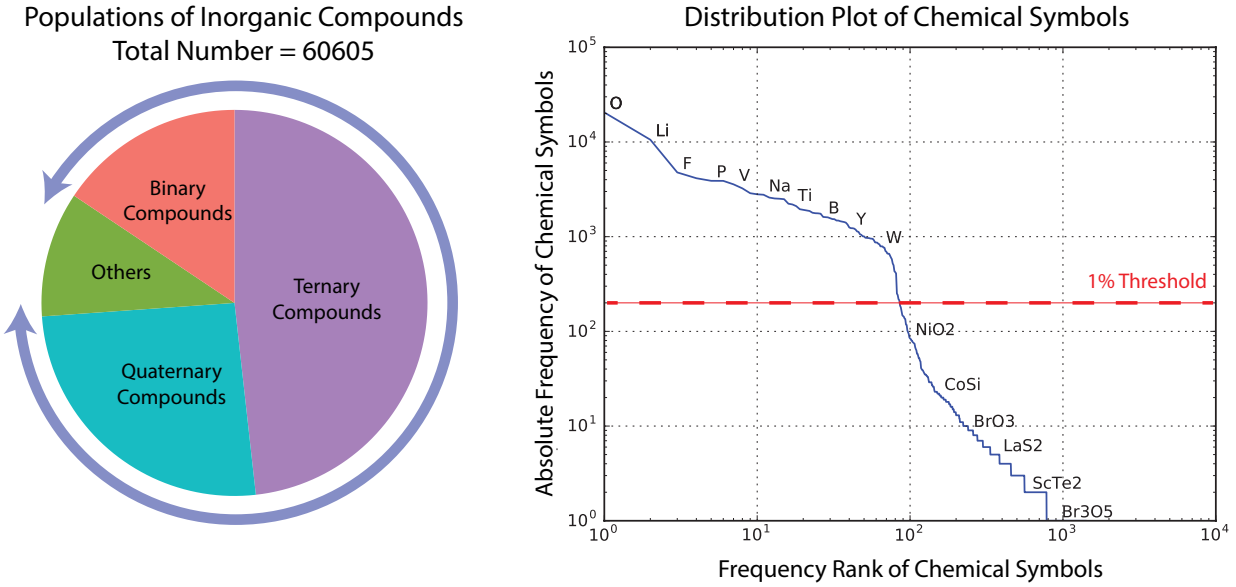

FIG. S1: (Color online) **Statistics of compound data for atom vector learning.** Population distributions of binary, ternary, quaternary and other compounds. Population plot of chemical symbols.

S2. Singular values in model-free method

Singular value decomposition[2] is applied on the atom-environment matrix in our model-free method. The singular values are shown in Fig. S2 in descending order. As seen in Fig. S2, a fat tail of non-vanishing values follows a quick decrease for the first ten, which indicates that even higher dimensions could describe meaningful aspects of atoms. In this work, atom vectors of dimension $d = 10, 20, 30$ are chosen as examples.

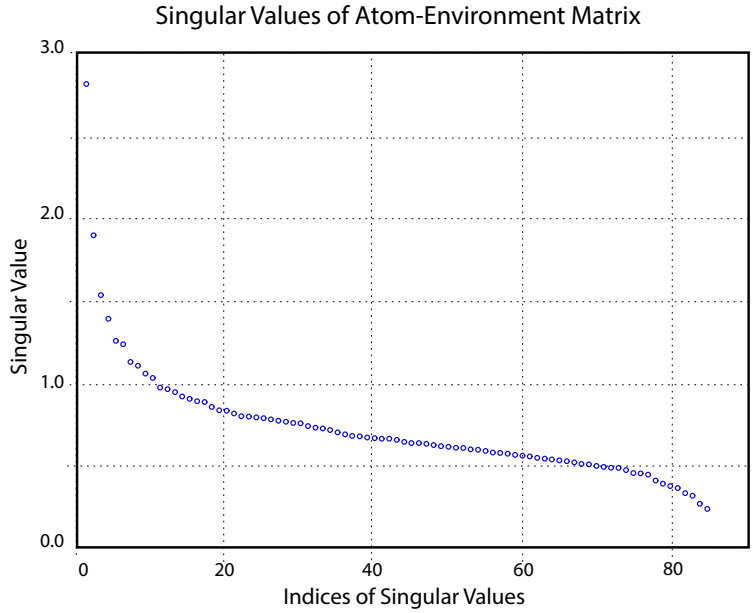

FIG. S2: (Color online) **Singular values distribution of atom-environment matrix.**

S3. Learning in model-based method

In model-based method, mini-batch stochastic gradient descent[3] is used to update atom vectors for minimization of the loss function. Fig. S3 shows learning curves for two atom vector dimensions ($d = 10$ and $d = 30$) based on two different score functions (bilinear and inverse square). When converged, the loss functions for all four cases are as high as 3.5, this means that the model itself still do not well describe the composition between atom and environment, since on average given an environment, the correctly predicted atom only gives probability of $e^{-3.5} \approx 1/30$.

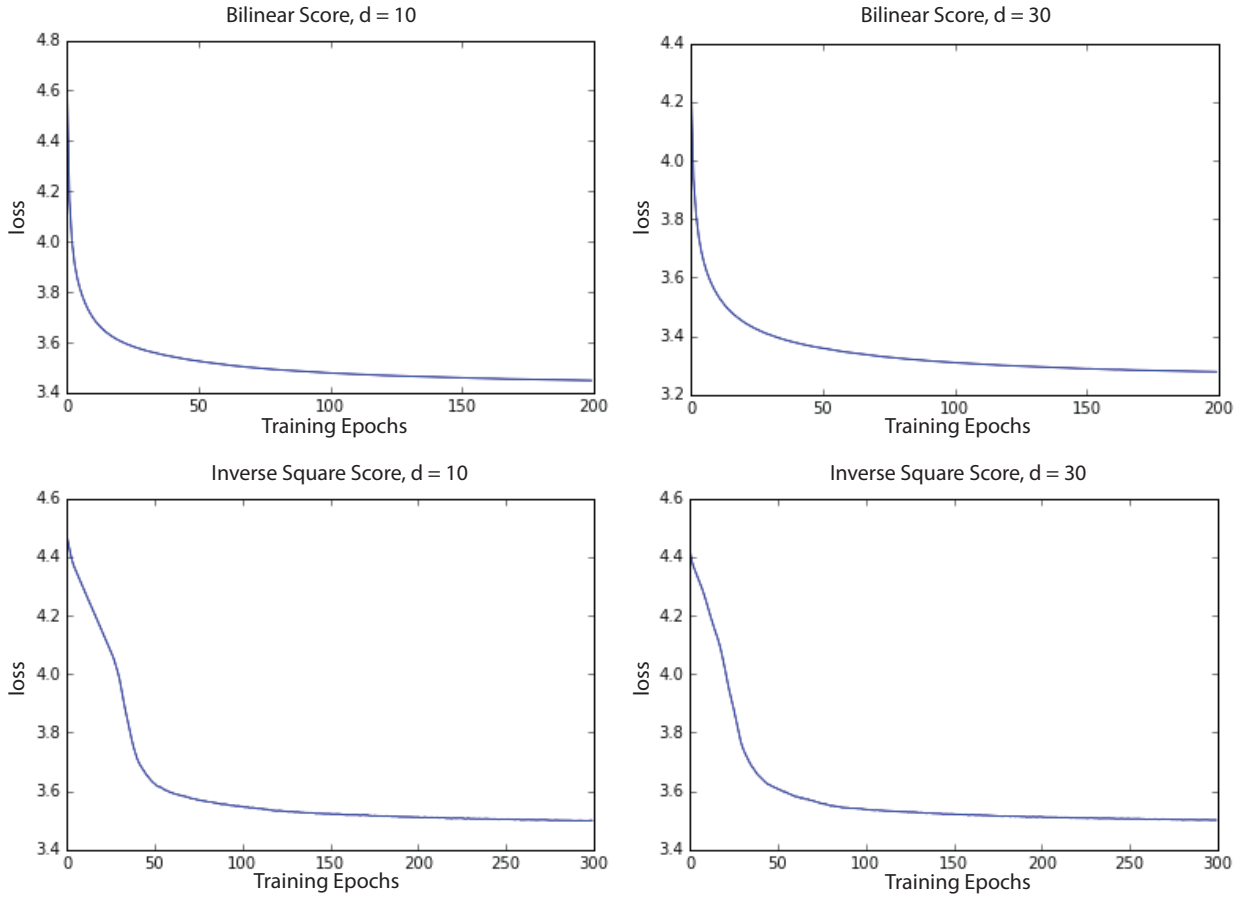

FIG. S3: (Color online) **Learning Curves for model-based methods** Two different score functions (bilinear and inverse square) and two atom vector dimension ($d = 10$ and $d = 30$) are investigated.

S4. Atom vectors from model-based method

We show atom vectors from our model-based method, in particular, from the one based on inverse square score function. The atom vectors of dimension $d = 20$ for main group elements are investigated, and projections to leading principal components are shown in Fig. S4. These vectors are learned from a small dataset including only compounds comprised of main group elements, rather than the entire dataset as in model-free method. We have also examined the case when the entire dataset is used, those vectors almost learn nothing about properties of atoms, probably due to the fact that the over-simplified model does not describe a large portion of the full dataset. This limitation actually appears in the atom vectors shown in Fig. S4 as well. Note that the first principal component of these vectors is dominant in model-based learning here, and it corresponds to the valence trend (or the columns in the periodic table) almost exactly. But projections over remaining components are extremely noisy, there appears no patterns at all beyond the first principal component in the model-based method.

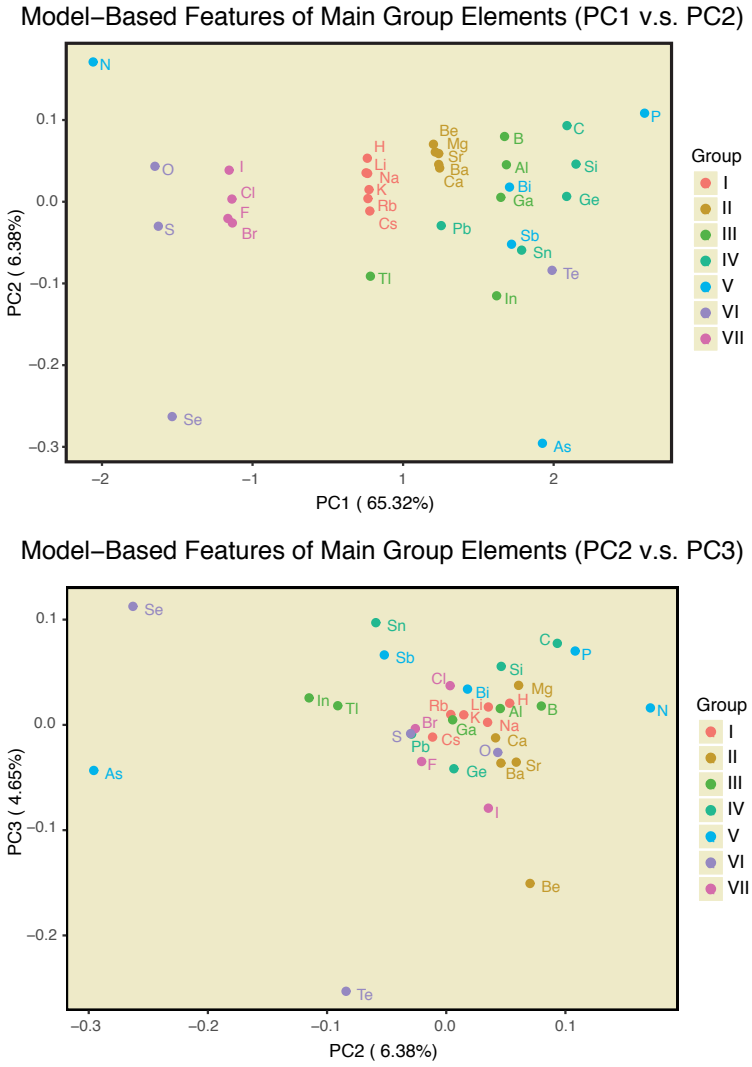

FIG. S4: (Color online) **Projection of atom vectors learned using model-based methods into leading principal components.** These atom vectors are of 20 dimensions, and inverse square score function is used in model-based learning. A small dataset which includes only compounds with atoms of main-group elements is used for learning.

S5. Elpasolite compound formation energy data

When evaluating our atom vectors quantitatively, we train neural network models to predict formation energy of elpasolites based on a dataset with formation energies computed by first principle calculation. There are nearly ten thousands of such compounds in the original dataset[4], only the compounds comprised of atoms that have our atom vector representation are used to train the model. This leaves about six thousands elpasolite samples, whose formation energy distribution is shown in Fig. S5.

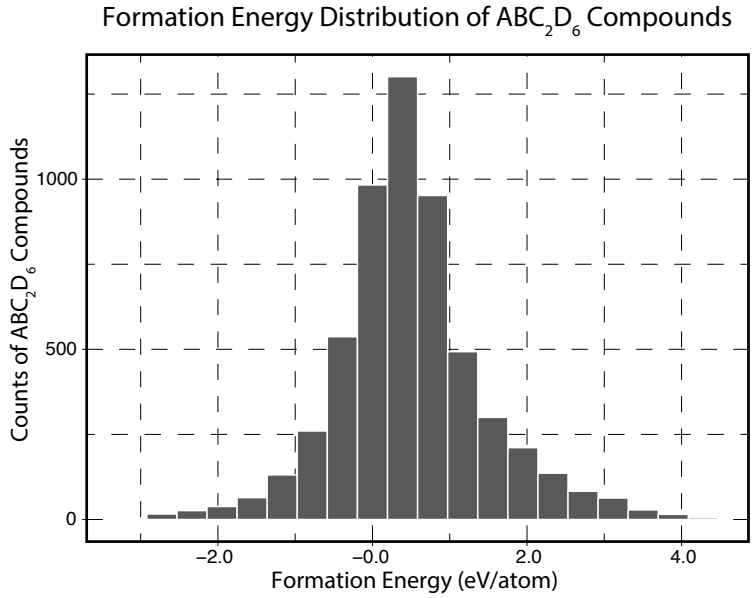

FIG. S5: (Color online) **Formation energy distribution of all ABC_2D_6 elpasolite compounds.**

References

1. Jain A, *et al*. (2013) The Materials Project: A materials genome approach to accelerating materials innovation. *APL Materials 1(1):011002*
2. Hastie T, Tibshirani R, Friedman J (2001) The Elements of Statistical Learning, Springer Series in Statistics. (Springer New York Inc., New York, NY, USA).
3. Goodfellow I, Bengio Y, Courville A (2016) Deep Learning. (MIT Press).
4. Faber FA, Lindmaa A, von Lilienfeld OA, Armiento R (2016) Machine learning energies of 2million elpasolite ABC2D6 crystals. *Phys. Rev. Lett 117(13):135502.*