CHEMISTRY OF MATERIALS

Article

pubs.acs.org/cm

Predicting the Thermodynamic Stability of Solids Combining Density Functional Theory and Machine Learning

Jonathan Schmidt,[†] Jingming Shi,[‡] Pedro Borlido,[§] Liming Chen,[∥] Silvana Botti,[§] and Miguel A. L. Marques*[,†]

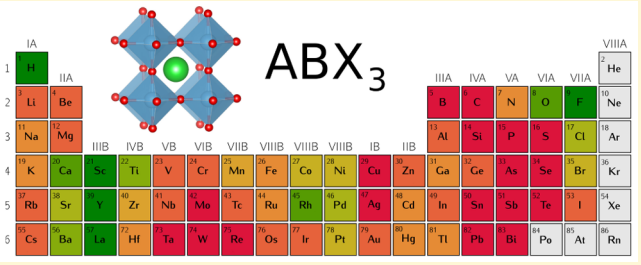
[†]Institut für Physik, Martin-Luther-Universität Halle-Wittenberg, D-06099 Halle, Germany

[‡]Université de Lyon, Institut Lumière Matière, UMR5306, Université Lyon 1-CNRS, 69622 Villeurbanne Cedex, France

[§]Institut für Festkörpertheorie und -optik, Friedrich-Schiller-Universität Jena and European Theoretical Spectroscopy Facility, Max-Wien-Platz 1, 07743 Jena, Germany

[∥]Liris laboratory UMR CNRS 5205, Ecole Centrale de Lyon, University of Lyon, 69134 Écully, France

ABSTRACT: We perform a large scale benchmark of machine learning methods for the prediction of the thermodynamic stability of solids. We start by constructing a data set that comprises density functional theory calculations of around 250000 cubic perovskite systems. This includes all possible perovskite and antiperovskite crystals that can be generated with elements from hydrogen to bismuth, excluding rare gases and lanthanides. Incidentally, these calculations already reveal a large number of systems (around 500) that are thermodynamically stable but that are not present in crystal structure databases. Moreover, some of these phases have unconventional compositions and define completely new families of perovskites. This data set is then used to train and test a series of machine learning algorithms to predict the energy distance to the convex hull of stability. In particular, we study the performance of ridge regression, random forests, extremely randomized trees (including adaptive boosting), and neural networks. We find that extremely randomized trees give the smallest mean absolute error of the distance to the convex hull (121 meV/atom) in the test set of 230000 perovskites, after being trained in 20000 samples. Surprisingly, the machine already works if we give it as sole input features the group and row in the periodic table of the three elements composing the perovskite. Moreover, we find that the prediction accuracy is not uniform across the periodic table, being worse for first-row elements and elements forming magnetic compounds. Our results suggest that machine learning can be used to speed up considerably (by at least a factor of 5) high-throughput DFT calculations, by restricting the space of relevant chemical compositions without degradation of the accuracy.



1. INTRODUCTION

In recent years there has been an increasing interest in the application of machine learning methods[1] to the fields of theoretical chemistry and solid-state physics. This was in part fueled by unparalleled advancements in other computational fields. In fact, machine learning techniques have now superhuman abilities (i.e., they perform better than an average human) in face recognition,[2,3] image geolocalization,[4] driving cars,[5] or even playing Go.[6]

Machine learning has already had a considerable success in the prediction of the properties of molecules[7,8] or polymers[9] and of dielectric properties,[10] in optimization of transition states,[11] and in creation of pair potentials for use in molecular dynamics simulations,[12,13] etc. For solids, one can also find applications to the determination of band gaps,[14−16] or to predict the stability of new compounds.[17−19] Also a considerable amount of work was performed to define requirements for suitable descriptors[20] or to find the best representation of a unit cell to be used as an input to machine learning algorithms.[21,22]

In this Article we will be concerned with solids and, in particular, with their stability and the prediction of new crystal phases. In this context, the most important material property is the free energy. This quantity determines, by itself, if a certain structure is thermodynamically stable or if it should decompose into other phases. This is particularly important in the new fields of high-throughput and accelerated materials design.[23] In these fields, one usually uses efficient and accurate numerical methods, such as density-functional theory (DFT), to identify promising compounds for technological applications. This approach has been applied, e.g., to battery materials,[24] superhard materials,[25] transparent conducting oxides,[26,27] perovskites,[28] organic polymers, dielectrics[29] and hard magnets,[30] to name just a few examples.

At this point there are still a series of open questions. For example, can machine learning methods be used to further

Received: January 13, 2017
Revised: May 15, 2017
Published: May 15, 2017

DOI: 10.1021/acs.chemmater.7b00156
Chem. Mater. 2017, 29, 5090−5103

accelerate the discovery of new materials, and by what margin? Which algorithms are better adapted to this task? What precision can we expect, and what is the typical size of the required training sets?

The only way to answer these questions convincingly is to perform large scale real-world benchmarks. Unfortunately, the amount of existing experimental data concerning energetics is rather limited (for the needs of machine learning), and it is often scattered, inconsistent, and difficult to obtain. To circumvent this problem, we decided therefore to develop an extensive synthetic benchmark. We used a rather common ternary stoichiometry (ABX_3) and a fixed crystal structure (cubic perovskite). By varying A, B, and X, and performing DFT calculations for every combination, we obtained a set of 250000 well-converged data points. This is one of the largest sets of DFT calculations for a fixed structure ever studied.

There are several reasons why we chose perovskites. These form a class of materials with the general formula ABX_3, where A is a 12-fold coordinated cation, B is an octahedrally coordinated cation, and X is an anion. They crystallize in a cubic crystal structure (see Figure 1) that is able to

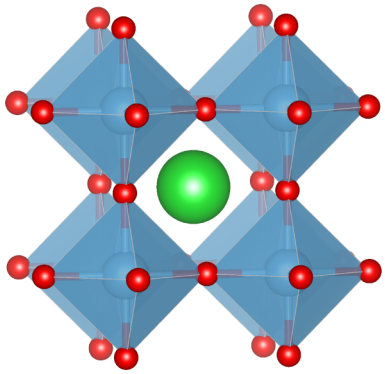


Figure 1. Perovskite structure with the 12-fold coordinated 1*a* (A atom in green) site and the octahedrally coordinated 1*b* site (B atom in blue). The X atoms in the 3*d* Wyckoff position are in red.

accommodate a large number of possible A, B, and X, with different oxidation states and sizes. Perovskites can occur naturally as a mineral (such as $CaTiO_3$), as well as synthetically fabricated [such as $Pb(Ti,Zr)O_3$]. This large variety also leads to a wealth of different material properties. In many cases, these properties are unmatched by any other known material, making perovskites the key for a variety of technologies essential to our modern society. A few examples include piezoelectrics, high-*k* dielectrics, superconductors, photovoltaic absorbers, and magnetoelectrics, etc.

This data set was then used to test several standard machine learning algorithms, namely, ridge regression, random forests, extremely randomized trees (including adaptive boosting), and neural networks. We find that extremely randomized trees give the smallest mean absolute error of the distance to the convex hull (121 meV/atom) in the test set of 230000 perovskites, after being trained with 20000 samples. As mentioned before, here we concentrate on total energies, but the described procedure is general and can be used to benchmark the prediction of lattice constants, band gaps, and so on.

Our analysis allows one to conclude that machine learning can be successfully combined with a learning machine to reduce the number of compounds to calculate by eliminating all systems that are safely far enough from the convex hull of stability.

The rest of this Article is organized as follows. In section 2 we present the details of the high-throughput calculations to generate the data set. Such a large set allows us, in section 3, to perform some interesting statistics concerning the physical properties of perovskites. Machine learning methods are introduced in section 4, and the results of the benchmark can be found in section 5. Finally, in section 6 we present our conclusions and a brief outlook on the future of machine learning to predict new stable materials.

2. HIGH THROUGHPUT

We start by building all possible ABX_3 compounds with a cubic perovskite structure (space group no. 221) with five atoms in the unit cell. All elements up to Bi, with the exception of the noble gases and the lanthanides, are taken into account. This amounts to 64 elements, leading to $64 \times 63 \times 62 = 249984$ different combinations. We then optimize the lattice constant and calculate the total energy, which can be done very efficiently due to the high symmetry of the cubic structure. To this end, we apply ab initio density-functional theory as implemented in the computer code VASP.[31,32]

All parameters were set to guarantee compatibility with the data available in the materials project database[33] and open quantum materials database.[34] Calculations were performed with spin polarization using the Perdew–Burke–Ernzerhof[35] (PBE) exchange-correlation functional, with the exception of oxides and fluorides containing Co, Cr, Fe, Mn, Mo, Ni, V, and W, where an on-site Coulomb repulsive interaction U with values of 3.32, 3.7, 5.3, 3.9, 4,38, 6.2, 3.25, and 6.2 eV, respectively, was added to correct the *d*-states. We used the PAW[36] data sets of version 5.2 with a cutoff of 520 eV and Γ-centered *k*-point grids, as dense as required to ensure an accuracy of 2 meV/atom in the total energy. All forces were converged to better than 0.005 eV/Å.

From the 249984 systems we managed to obtain results for 249654, while the remaining 330 (0.13%) failed to converge in spite of our efforts. A more careful look indicates that these are often highly unstable phases, which are consistently predicted at a large distance from the convex hull by the machine learning algorithm and which are therefore irrelevant for our analysis. A final remark on Cs: it turns out that its pseudopotential of version 5.2 of VASP leads often to numerical problems with spin-polarized calculations. In these cases, and as an ultimate measure to circumvent the low quality of this pseudopotential, we resorted to spin-unpolarized calculations.

We worked always at zero temperature and pressure, and we neglected the effects of the zero-point motion of the phonons (that are expected to be negligible for these ternaries that are unlikely to contain only light atoms, as we will discuss later). In this case, the free energy is simply given by the total energy of the system. With this quantity we can then evaluate the convex hull of stability and therefore the energy distance of a compound to the convex hull, which gives us a direct measure of its (in)stability. The convex hull is a (hyper)surface of the formation energy as a function of the chemical composition that passes through all lowest energy phases that are "thermodynamically stable", i.e., that do not decompose (in a possibly infinite time) into other phases. In other words, a material is thermodynamically stable if its free energy is lower than the free energy of all possible decomposition channels, including not only elementary substances but also other

DOI: 10.1021/acs.chemmater.7b00156
Chem. Mater. 2017, 29, 5090–5103

possible binary, ternary, and so on (reservoir) compounds. We emphasize that the energy distance to the convex hull gives us a much better measure of stability than, e.g., the cohesive energy that only takes into account the possible decomposition into elementary substances.

To evaluate the convex hull, one needs the total energy of all possible reservoir compounds. Fortunately, this information is already available in excellent public databases, such as the materials project,[33] open quantum materials database,[34] and the ab initio electronic structure library AFLOWLIB.[37] We chose to use the materials project database for our reference energies and to determine the distances to the convex hull of stability with PYMATGEN.[38] The materials project database includes most of the experimentally known inorganic crystals that are present in the ICSD database[39,40] and an increasing number of theoretically predicted phases.

As we will see in the following, we will be performing regression using machine learning. As such, and to have a smoother function to predict, we decided to intentionally exclude all compositions of the type ABX_3 to construct the convex hull. Therefore, "stable" cubic perovskite ABX_3 compounds will appear as having a *negative* energy distance to the convex hull. The larger (in magnitude) negative distance, the more energy will be required for the compound to decompose.

3. DATA SET

In Figure 2 we look at the energy distance to the convex hull for the materials of composition ABX_3 present in ICSD that

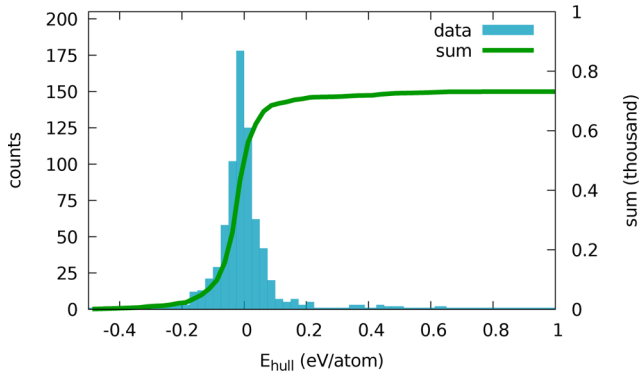


Figure 2. Histogram of the distribution of E_{hull} for all the structures contained in the materials project database with an ICSD number. The bin size is 25 meV/atom.

were calculated in the Materials Project. Note that this contains not only perovskites but also all other known crystal structures for this composition, in a total of 736 different stoichiometries and 1616 materials.

Figure 2 helps us to estimate the reliability of our calculations, and more specifically the fraction of false negatives that we have to account for. As expected a large majority of materials turns out to have a negative or small positive energy distance to the convex hull within the PBE approximation. We find 436 materials (59%) with negative distances to the convex hull; considering distances below 25 meV/atom we count 561 compounds (76%), while 626 have a distance below 50 meV (85%). To have 95% of all structures present in ICSD, we have to go to around 150 meV above the hull.

At this point we need to clarify a few points: (i) It is sometimes possible to experimentally synthesize a material with a positive distance to the hull, as it can be stabilized by vacancies, defects, and temperature, etc. (ii) We could find some obvious errors in several database entries that translate into large distances to the hull, deteriorating the statistics. For example, for the perovskite compounds $ScCIr_3$, $AuNV_3$, $InCMn_3$, $PbNCa_3$, and $SnNCa_3$, the A and B atoms seem to be interchanged in the database. (iii) Many of the database compounds with large distances to the convex hull are actually not perovskites, such as NOK_3, $InPF_3$, and CPF_3, etc. This means that the numbers that we have presented above are a pessimistic picture of reality.

In Figure 3 we plot the histogram of the energy distances to the convex hull for all ≈250000 cubic perovskite structures

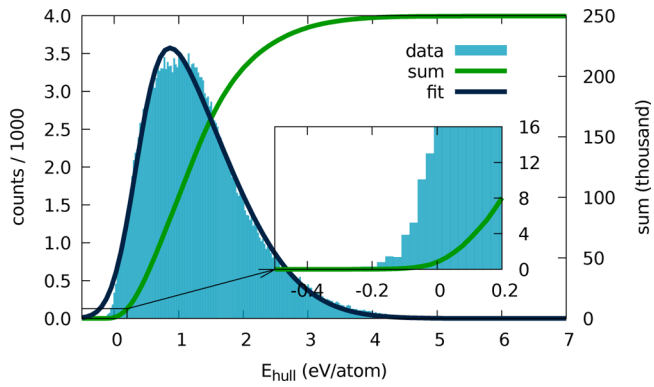


Figure 3. Histogram of the distribution of E_{hull} for all ≈250000 cubic perovskite structures. The bin size is 25 meV/atom.

from our high-throughput study. The distribution is very smooth, with no clear boundary between stable and unstable compounds. The curve can be fitted with a skew normal distribution given by

$$f(x) = \frac{2N}{\sigma\sqrt{2\pi}} e^{-x^2/2} \left\{ \frac{1}{2} [1 + \mathrm{erf}(x\,\alpha/\sqrt{2.0})] \right\} \quad (1)$$

where N is the number of structures, $x = (E_{hull} - E_0)/\sigma$, $E_0 = 0.351 \pm 0.005$ eV/atom, $\sigma = 1.206 \pm 0.009$ eV/atom, and $\alpha = 3.8 \pm 0.1$. We do not have any theoretical justification for the use of this curve, but we can see that such a simple form is able to reproduce quite well the data (see Figure 3). Note that a log-normal distribution can also fit equally well the data. From the figure it is clear that there are less stable compounds than those predicted by a Gaussian tail, even if the tail for large energies seems to decay slower than the fitting function.

The number of compounds increases very rapidly with the energy distance to the convex hull. There are 641 formally stable compounds within our theoretical framework (i.e., with $E_{hull} \leq 0$), while there are 699 below 5 meV/atom, 1015 below 25 meV/atom, 1512 below 50 meV/atom, and 2923 below 100 meV/atom. We will define as "stable", from now on, all compounds with an energy distance from the convex hull E_{hull} smaller than 5 meV/atom.

In Figure 4 we represent the number of stable structures for every element in the 1*a*, 1*b*, and 3*d* positions. There are some obvious trends in the plots. In the position 1*a* we find that the most favorable elements are Cs and Tl, with the probability of finding stable perovskites decaying slowly as we move from these elements. Very few stable systems are found with non-

DOI: 10.1021/acs.chemmater.7b00156
Chem. Mater. 2017, 29, 5090−5103

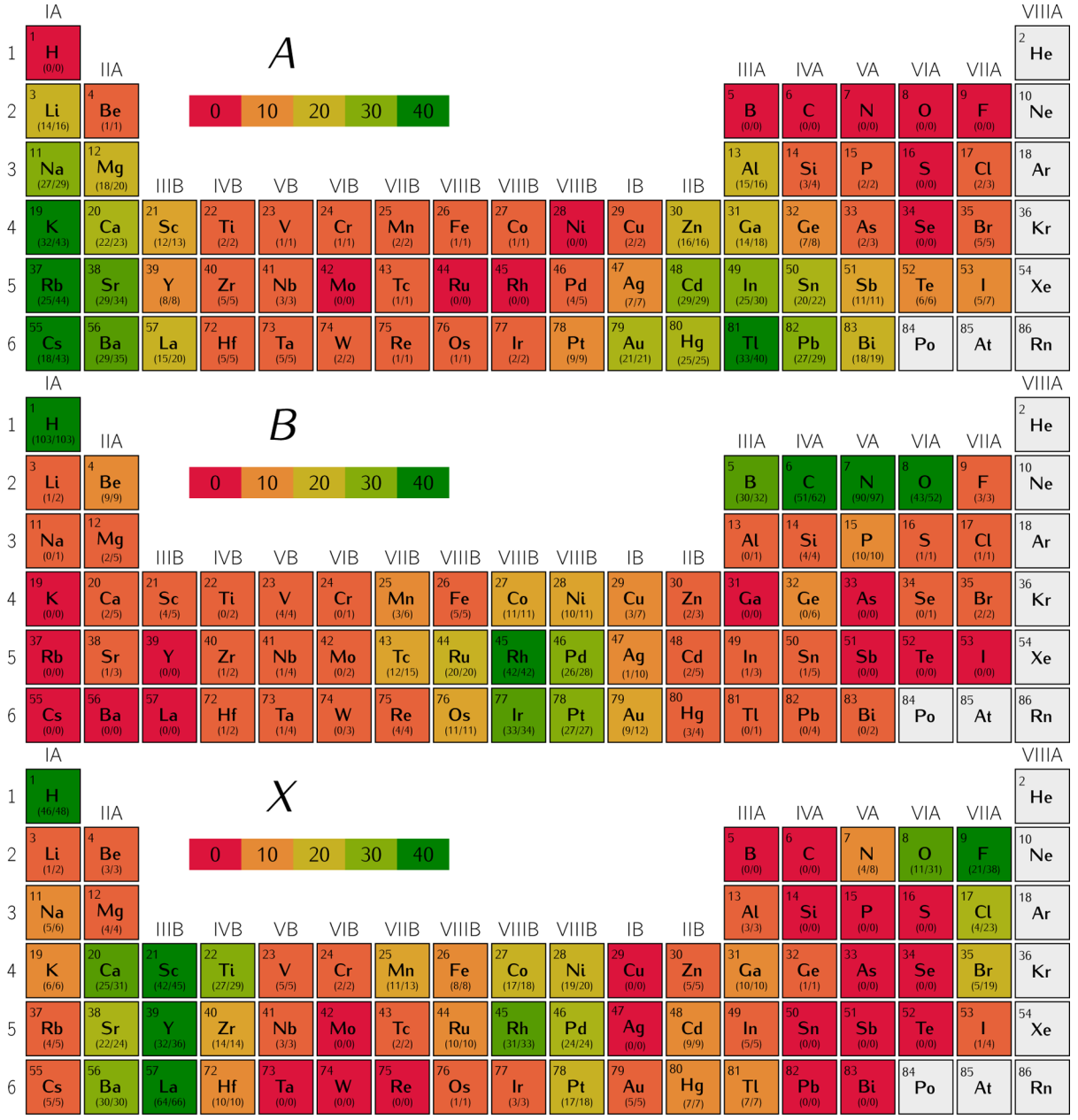


Figure 4. Periodic tables showing the stable (E_{hull} < 5 meV/atom) structures for every element A, B, and X, respectively, in the 1*a*, 1*b*, and 3*d* positions. The numbers in parentheses below each symbol represent the total number of new stable structures (i.e., that are not already present in the materials project database) and the total number of stable structures.

metals, transition metals, or light elements in this position. Position 1*b*, on the other hand, favors light elements such as H, B, C, N, and O, although a number of transition metals (especially around Rh) can also be found to stabilize the perovskite structure. Finally, for position 3*d* the situation is more complicated. On the right side of the periodic table we see that stability increases as we go to the upper right, with a trend that follows essentially increasing electronegativity. On the left side, both hydrogen and metals around Sc form several stable perovskites. Finally there is an island of stability for transition metals of the group VIIIB.

From Figure 4 we can also see that there is an enormous amount of new inverted perovskite systems, i.e., systems with a metal in the 3*d* Wyckoff position, not present in the materials project database. For example, for Sc we find 36 new phases (out of 45), for Y we find 30 new phases (out of 26), and for La 62 (out of 67), etc. Furthermore, we discover several new "exotic" stable perovskite families, for which no experimental system was found in the databases. This is true for Be, Mg, Zr, Hf, V, Zn, and Ga, etc.

A final word regarding the so-called inter-metallic perovskites. This term is usually used for non-oxide systems, and in particular borides and carbides[41] (such as $MgCNi_3$, $GaCMn_3$, $ZnCMn_3$, and $SnCMn_3$, etc.). These are very interesting materials, as they exhibit superconductivity[42,43] and magnetism,[44,45] and they can be used to strengthen aluminum-alloyed steels.[46] We do find a large number of new carbide perovskites (51 systems), such as $SbCSc_3$, $AuCTi_3$, $LiCNi_3$, and $SnCLa_3$, etc. Moreover, we find a series of other new truly inter-metallic

DOI: 10.1021/acs.chemmater.7b00156
Chem. Mater. 2017, 29, 5090−5103

systems not containing boron or carbon, such as $LaBeGe_3$, $GeReGa_3$, $BaIrZn_3$, and $SrRhCa_3$, etc.

In Figure 5 we show a histogram of the formation energy. This curve is centered at 0.81 eV, has a standard deviation of

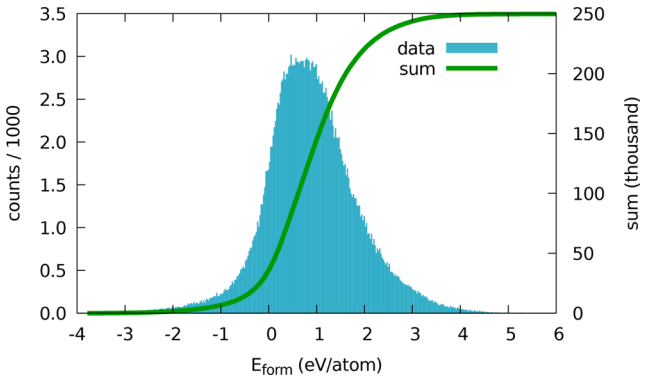


Figure 5. Histogram of the distribution of E_{form} for all ≈250000 cubic perovskite structures. The bin size is 25 meV/atom.

0.65 eV, and is also slightly skewed toward higher energies. It is also clear that a large majority of the compositions has a positive formation energy; i.e., the corresponding perovskites are unstable toward decomposition into elementary substances. Up to recently, the formation energy was used extensively in the literature to study the stability of compounds. Comparing the results in Figure 5 with Figure 3, we immediately realize that this is a dangerous approach. In fact, there are 35028 materials with $E_{form} < 0$, while there are only 641 with a negative energy distance to the convex hull. The difference amounts to the number of materials that do not decompose to elementary substances, but to binary or other ternary compositions.

In Figure 6 we plot the distribution of minimum (indirect or direct) band gaps. Note that these band gaps are calculated with

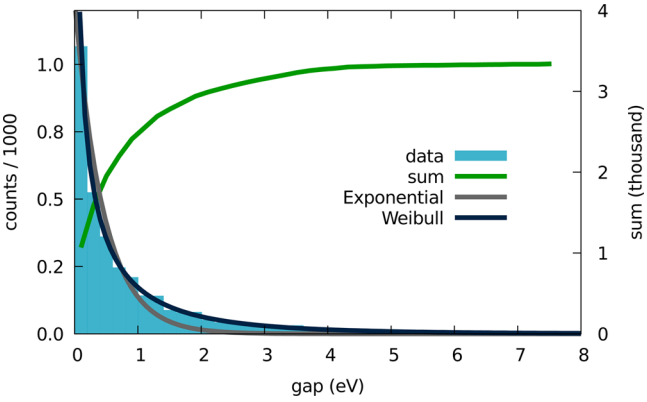


Figure 6. Histogram of the distribution of the minimum band gap for all ≈250000 cubic perovskite structures. The bin size is 25 meV/atom.

the PBE approximation to the exchange-correlation functional and are therefore considerably underestimated. Moreover, around half of all compounds are necessarily metallic as they have an odd number of electrons in the primitive five-atom cubic cell. Despite these limitations, it is possible to extract some trends from these results. The first thing we notice is that there is a very small number of semiconducting systems. Only 3340 have a nonzero gap, and from these only 1562 have a gap larger than 0.5 eV. We naively expected that the curve of Figure 6 would decay exponentially, but in reality the distribution of gaps has a rather fat tail. It can be very well described by a two-parameter Weibull distribution of the form

$$w(x) = N\frac{k}{l}\left(\frac{x}{l}\right)^{k-1} e^{-(x/l)^k} \quad (2)$$

where $N = 749 \pm 10$ is a normalization parameter and $k = l = 0.65 \pm 0.01$. Of course we presume that these parameters will change if the experimental gap, or a more accurate prediction of the gap, could be plotted instead of the PBE gap. If this behavior is specific to the perovskite system, or is a universal phenomenon across different crystal structures, remains at the moment an open question.

We also looked at the relationship between band gap and stability. It is well-known for molecules that the presence of a large gap between the highest occupied state and the lowest unoccupied state is usually associated with stability. In fact, one expects that molecules with small gaps distort through a Jahn–Teller mechanism to give more stable structures with larger gaps.[47] This observation eventually led to the definition of the maximum hardness principle,[48] which states that a system will stabilize by maximizing its chemical hardness (a quantity directly related to the gap). This effect has also been studied for solids,[49] although the distortion route is often frustrated in solids (metals do exist experimentally!).

In our calculation we fixed the crystal symmetry to the cubic perovskite structure, so no distortion is allowed. However, we can see that there is still a clear correlation between the size of the band gap and the stability. In Figure 7 we show a scatter

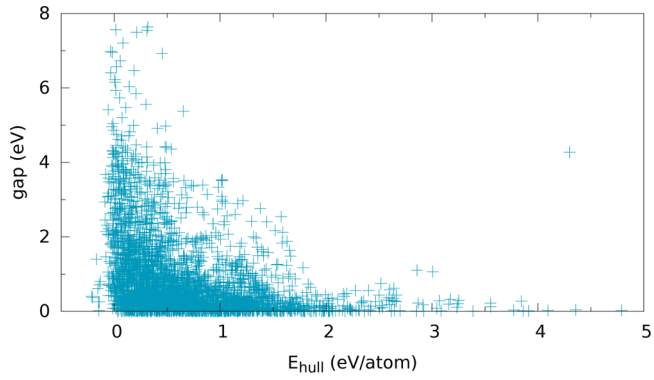


Figure 7. Scatter plot of the band gap versus energy distance to the convex hull of stability. Each point corresponds to a semiconducting phase.

plot of the band gap versus energy distance to the convex hull. We observe that the systems with the largest gaps are significantly more likely to have negative or smaller positive distances to the convex hull of stability.

We plot in Figure 8 the number of valence electrons for all the structures studied here and for the stable perovskites ($E_{hull} < 5$ meV/atom). The double-peak structure in the blue histogram is due to the fact that we did not include lanthanides in our study. We can find stable perovskites in a large range of number of valence electrons, from the six valence electrons of $BaLiH_3$, $SrLiH_3$, and $KMgH_3$ to the 48 of $BaPtZn_3$. This is in striking contrast with half-Heusler ABX compounds, where nearly all stable materials have either eight or 18 valence

DOI: 10.1021/acs.chemmater.7b00156
Chem. Mater. 2017, 29, 5090−5103

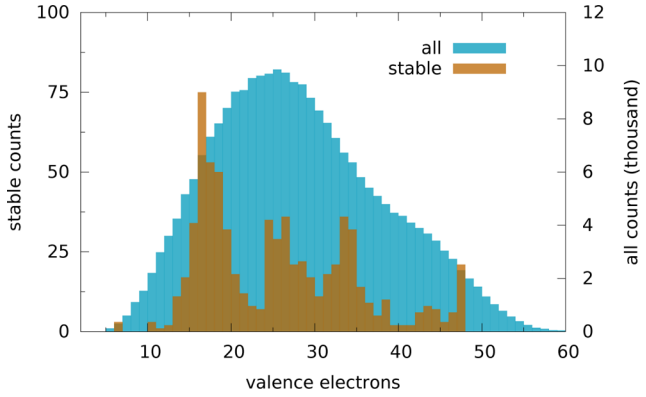


Figure 8. Histogram of the number of valence electrons for the stable structures (E_{hull} < 5 meV/atom).

electrons.[50] We also do not find a marked difference between odd and even numbers of electrons.

To conclude our preliminary analysis of our set of DFT calculations, in Figure 9 we show the distribution of the PBE

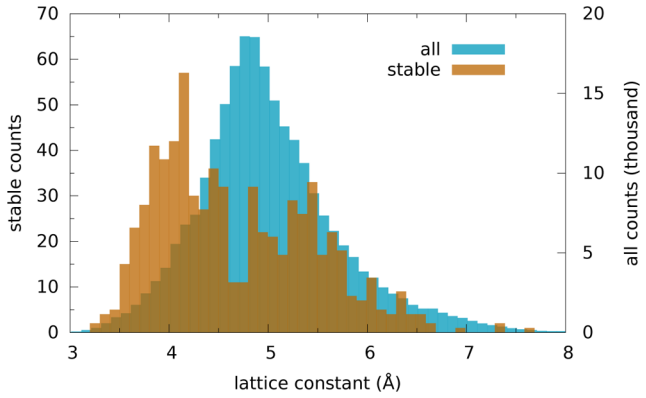


Figure 9. Histogram of the lattice constant for all and for the stable structures (E_{hull} < 5 meV/atom). The width of the bin is 0.1 Å.

optimized lattice constant for all structures or only for the stable structures. For our chosen set of elements, the average perovskite structure has a lattice parameter a = 4.89 pm ± 0.01 Å, with a standard deviation of σ = 0.60 ± 0.01 Å. The stable perovskites do not follow the same distribution, tending clearly to smaller lattice constants. This can be easily understood if we keep in mind the marked preference for light elements in the Wyckoff 1*b* position (see Figure 4).

Finally, we remark that within our set of stable cubic perovskites we find 97 compounds with a nonzero magnetic moment. The maximum magnetic moment is 8.9 Bohr magnetons per unit formula for $GaHMn_3$, but we can find several other manganites and cobaltites with rather large magnetic moments (>5 Bohr magnetons). Among the magnetic systems we find 21 with Mn, 11 with Fe, 14 with Co, 10 with Ni, and 3 with Cr.

The crystal structure of all stable perovskites will be distributed through the Materials Project database,[33] while the whole data set of cubic perovskites will be available in the NOMAD Repository.[51]

4. MACHINE LEARNING METHODS

In this section, we give a very brief description of the machine learning algorithms that we used to predict the stability of the cubic perovskite systems. We note that by deciding on a specific machine learning algorithm, one determines the model the computer uses. More sophisticated models may allow for better predictions, but usually require more data for their training.

Our problem falls into the category of supervised learning. We could decide to handle it in two different ways: (i) as a classification problem, where the machine should simply predict if a compound is stable (1) or not (0), or (ii) as a regression problem, where the machine predicts the energy distance to the hull. We experimented with both but could not find any advantage in using classification algorithms. Moreover, the actual value of E_{hull} has a physical meaning that can be used in the interpretation of the results, so it is advantageous to use a method that returns it. Therefore, in the following we restrict our discussion to regression.

Given a set of N training samples of the form $\{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$ such that $\mathbf{x}_i$ is the input feature vector (also called descriptor) of the ith element and y_i is its distance to the convex hull E_{hull}, the selected learning algorithm must seek the function $y(\mathbf{x}_i)$ that best fits the data. In order to measure how well a function fits the training data, a loss function is defined and minimized.

4.1. Ridge Regression. The standard approach for linear regression is finding a ridge regression estimate vector $\boldsymbol{\beta}$ that minimizes the squared error loss function $\mathcal{L} = \sum_i (y_i - \mathbf{x}_i^T\boldsymbol{\beta})^2$. Unfortunately, this method has several disadvantages. If the features are correlated, the solution has a high variance, and if they are linearly dependent, the normal equation has no solution. In order to counteract this behavior, and to favor one particular solution, one can include a regularization term in the loss function. This method is called Tikhonov regularization or ridge regression. We chose the Tikhonov matrix $\boldsymbol{\Gamma}$ as $\boldsymbol{\Gamma} = \lambda\mathbf{I}$ (also known as L_2 regularization), leading to

$$\mathcal{L} = \sum_i (y_i - \mathbf{x}_i^T\boldsymbol{\beta})^2 + \lambda \sum_j \beta_j^2 \tag{3}$$

This favors solutions with a smaller L_2 norm.[52]

Ridge regression is one of the most straightforward methods of machine learning, quite robust and very easy to use. However, the underlying model is rather simple, and therefore one cannot expect it to be a good predictor. We include it in our analysis mainly to have a comparison with the other, more sophisticated methods.

4.2. Decision Trees, Random Forests, And Extremely Randomized Trees. In broad terms, a decision tree is a graph-like structure in tree form.[53] Classically when searching for the best logic condition C_i, one uses a metric to determine the attribute and the splitting point at each node, e.g., the Gini impurity[54] or an information gain.[55] Unfortunately, these classic methods are very prone to overfitting.

Random forests are a method to reduce the variance of decision trees. A random forest regressor[56] is an ensemble of decision trees $\{h(\mathbf{x}_i, \Theta_k), k = 1, ...\}$. The Θ_k are independent random vectors which were used to randomize the tree-building process. In this implementation of random forests the random vector Θ_k is used for bootstrap aggregating[57] and picking random features to split a node. Given a training set, bootstrap aggregating (also called bagging) generates new training subsets by picking random samples with replacement out of the original set, meaning that a sample can be picked multiple times. Each new set is then used to train an individual tree. When searching for the best possible split, a random

DOI: 10.1021/acs.chemmater.7b00156
Chem. Mater. 2017, 29, 5090−5103

selection of features is chosen at each node. When training a random forest for regression, one typically picks one-third of the features. The split is chosen by minimizing the mean squared error which corresponds to a variance minimization.[58] In the end, the result for an input vector $\mathbf{x}_i$ is the average of all predictions.

Extremely randomized trees[59] is another algorithm to reduce the variance of decision trees, but it differs in several ways from random forests. In this work the ExtraTree[59] algorithm was used in its classic version. The ExtraTree algorithm does not use bootstrap aggregating, but it randomizes the tree growing process by picking random splitting points. More specifically, at each node N random attributes a_i are selected, and for each of them a random cut-point $a_{i,c} \in [a_{i\min}, a_{i\max}]$ is drawn. The set of samples at each node is split into two parts by the binary condition $a_i < a_{i,c}$, and the Gini impurity (mean squared error for regressors) is computed to evaluate the splits. Afterward the split with the best score is chosen. The node splitting process is repeated until one of the following conditions applies: (i) the set is smaller than the minimum number of samples per set; (ii) all samples in the set have the same value for all attributes; (iii) all samples in the set result in the same output. This procedure is repeated to produce an ensemble of trees. The output of the ensemble is set to the mean of the single-tree outputs.

4.3. Neural Networks. In general, neural networks take inputs and connect them in a nonlinear fashion to calculate an output. Every neural network consists of at least one input layer and one output layer. Between those layers can be an arbitrary number of hidden layers which transform the input values. Each layer consists of a number of neurons, which can be represented by a vector $\mathbf{z} \in \mathbb{R}^n$, with n being the number of neurons in the layer. $\mathbf{z}_i$ represents the ith layer in a neural network, and z_{ij} is the jth neuron in the ith layer with $j \in \{0, 1, ..., n_i\}$.

When one wants to compute the output of a neural network for a sample, the input layer $\mathbf{z}_0$ is set equal to the feature vector representing the sample. The n_j neurons in the ith layer are connected to the previous layer through activation functions φ

$$z_{ij} = \varphi_i\left(\sum_k^{n_i} \theta_{jk} z_{i-1,k}\right) \tag{4}$$

where θ_{jk} are a set of weights to be obtained through the training procedure. The network's designer decides which activation functions φ_i are used.

The original idea behind neural networks was to emulate networks of biological neurons in a computer. The firing rate of a biological neuron stays close to zero as long as no input is received. Once an input is received, the firing rate first rises quickly and then approaches asymptotically 100%. Classically normalized sigmoid functions have been the most popular activation functions because they are a simple way to represent the firing potential of a biological neuron. All sigmoid functions fulfill these requirements, but typically the logistic function and the hyperbolic tangent were used.

$$\varphi_{\text{logistic}}(x) = \frac{1}{1 + e^{-x}} \tag{5a}$$

$$\varphi_{\tanh}(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{5b}$$

The logistic function has several disadvantages. Since its gradient is $f'(x) = f(x)[1 - f(x)]$, which is a vanishing quantity for $f(x)$ close to 0 or 1, the logistic function becomes very easily saturated. Once the upper layers of the network are close to being saturated, they back-propagate near-zero gradients. This issue is known as the vanishing gradient problem[60] and either can cause slow convergence or result in poor local minima. Second, the output of sigmoid functions is non-zero-centered which results in slower convergence. On top of that, the exponential function is relatively expensive to compute. (There are however approximations such as the hard sigmoid which can be used to lower the computational time.) Compared to the logistic function, the hyperbolic tangent has the advantage of being zero-centered.[61]

With the introduction of rectified linear units (ReLU)

$$\varphi_{\text{ReLU}}(x) = \max(0, x) \tag{6}$$

many of these problems were solved.[62] They are extremely easy to compute, and as long as x stays positive, they do not saturate, meaning that their gradient does not approach zero as easily. In addition, neural networks with ReLU activation functions converge several times faster than similar networks with sigmoid functions.[62] Besides their output being non-zero-centered, the other disadvantage of ReLU activation functions is that they become saturated as soon as x becomes smaller than or equal to zero. To bypass this problem, one can use leaky ReLUs[63,64] which do not saturate:

$$\varphi_{\text{leaky ReLU}}(x) = \max(0.001x, x) \tag{7}$$

When training a neural network for classification or regression, the last layer is usually a loss layer $\mathcal{L}$ which computes a measure of error for the output of the final layer $\mathbf{z}_{\text{out}}(\mathbf{x}_i)$ by comparing it to the training values $\mathbf{y}_i$. In this work, the loss $\mathcal{L}$ was computed using a simple square loss function:

$$\mathcal{L} = \sum_i (\mathbf{z}_{\text{out}}(\mathbf{x}_i) - \mathbf{y}_i)^2 \tag{8}$$

In order to minimize the loss function, the weight matrices of the neural networks are trained with back-propagation. When back-propagating, the partial derivative of the loss function with respect to the weights is computed. The derivative is computed by using the chain rule; therefore all activation functions must be differentiable. Steepest descent or other more sophisticated methods are used to find the minimum of the loss function.

In order to reduce overfitting during the training process, a regularization term containing the norm of the weights can be added to the loss function:

$$\mathcal{L}_{L_1} = \lambda \sum_{ijk} |w_{ijk}| \tag{9}$$

This L_1 regularization causes the error to be smaller when using smaller weights, therefore giving preference to simpler hypothesis.

4.4. Adaptive Boosting. Adaptive boosting (AdaBoost)[65] produces ensembles of machines in order to reduce their variance compared to the single machine. Therefore, it has to be used in conjunction with another machine learning algorithm. The boosting algorithm picks a training subset and uses the main machine learning algorithm to build regressors (or classifiers). Over the course of the boosting process the probability of each sample to be picked for training is changed as a function of the relative errors the regressor produces for that sample.

DOI: 10.1021/acs.chemmater.7b00156
Chem. Mater. 2017, 29, 5090−5103

In order to calculate the error, different loss functions can be used as long as $\mathcal{L}^j \in [0, 1]$, such as[65]

$$\mathcal{L}^{i,j}_{\text{linear}} = \frac{|y_j(\mathbf{x}_i) - y_i|}{\sup_i |y_j(\mathbf{x}_i) - y_i|} \tag{10a}$$

$$\mathcal{L}^{i,j}_{\text{square}} = \frac{|y_j(\mathbf{x}_i) - y_i|^2}{\sup_i |y_j(\mathbf{x}_i) - y_i|^2} \tag{10b}$$

$$\mathcal{L}^{i,j}_{\text{exp}} = 1 - \exp\left[-\frac{|y_j(\mathbf{x}_i) - y_i|}{\sup_i |y_j(\mathbf{x}_i) - y_i|}\right] \tag{10c}$$

where j denotes the machine and i the sample.

Samples with a high relative error have an increased likelihood of appearing in the training set so more time is spent training regressors on the difficult samples. After training, the results of the regressor ensemble are combined to boost the precision of the cumulative prediction. When combining the different regressors, the more confident regressors are weighted more heavily.

In the following AdaBoost as proposed in ref 65, which is in turn a modification of AdaBoost.R,[66] will be explained. In every step of AdaBoost one regressor is trained on a subset of the training set. This subset is determined by picking N samples with replacement randomly out of the original training set. Initially, each training sample is assigned the same weight $w_i = 1$ and thus the same initial probability

$$p_i = \frac{w_i}{\sum_i w_i} \tag{11}$$

of being picked for training. Then a random subset of N training samples is picked with replacement. These samples are used to build a regressor y_j, which is in turn used to calculate a prediction $y_j(\mathbf{x}_i)$ for every sample $\mathbf{x}_i$ in the whole training set. The average loss

$$\overline{\mathcal{L}^j} = \sum_{i=1} \mathcal{L}^j_i p_i \tag{12}$$

is then calculated in order to introduce β:

$$\beta_j = \frac{\overline{\mathcal{L}^j}}{1 - \overline{\mathcal{L}^j}} \tag{13}$$

which is a measure of confidence in the predictor, where a low β corresponds to a confident prediction by the regressor.

After calculating β the weights w_i are updated as $w_i^{\text{new}} = w_i \beta_j^{[1-\mathcal{L}^j_i]}$. When the prediction is accurate, the weight of the sample is reduced. A small loss $\mathcal{L}^j_i$ will also reduce the weight of the sample thus reducing the probability of the sample being picked again.

The combined prediction $y(\mathbf{x}_i)$ of the boosted regressors for an input $\mathbf{x}_i$ is found by calculating a weighted median. For this all regressors make a prediction $y_j(\mathbf{x}_i)$, which is sorted so that $y_1(\mathbf{x}_i) < y_2(\mathbf{x}_i) < y_3(\mathbf{x}_i) < \ldots < y_N(\mathbf{x}_i)$. The indices of the β_j are changed accordingly keeping the connection to the $y_j(\mathbf{x}_i)$ intact. Every prediction is weighted by $\log \frac{1}{\beta_j}$, and the weighted median is calculated so that $y(\mathbf{x}_i)$ is the smallest y_t for which the sum of the weights of the previous y_j are larger than or equal to half the total weight:

$$y(x_i) = \inf\left\{ y_t : \sum_{j: y_j < y_t} \log \frac{1}{\beta_j} \geq \frac{1}{2} \sum_j \log \frac{1}{\beta_j} \right\} \tag{14}$$

5. BENCHMARK

To perform the benchmarks with the algorithms described in section 4, we used the implementations from SCIKIT-LEARN[67] (for decision trees and boosting), CAFFE[68] (for neural networks), and our own implementation for ridge regression.

All compounds with an energy distance to the convex hull smaller than −0.5 eV/atom or larger than 3 eV/atom were treated as outliers in the DFT calculations and were removed from the training and the test set. Our data set contains exactly 249692 compounds, of which one sample has a distance to the convex hull smaller than −0.5 eV/atom and 9642 samples have a distance to the convex hull larger than 3 eV/atom, thereby representing 0.00% and 3.86% of the whole data set, respectively. From the material science point of view, the compounds above 3 eV are totally uninteresting as they are highly unstable. On the other hand, the only large negative distance to the known convex hull of stability, i.e., less than −0.5 eV, is likely due to an incomplete knowledge of the hull around that composition. As a result, in order to gain prediction precision in our simulations, we have decided to remove these 9643 compounds from the data set for training and testing. In the following, machine learning algorithms were trained using 20000 randomly chosen compounds, while the rest was used for testing (unless stated otherwise). In order to measure the quality of the regressors, the mean absolute error (MAE) of the test set was used.

We used as an input for our machines a maximum of 119 input features ranging from basic atomic information to physical properties of the elementary substance. More specifically, we used for every element the number of valence electrons; the empirical value of the atomic number; Pauling electronegativity and the difference of Pauling electronegativities; maximum and minimum oxidation states; all common oxidation states; atomic mass; atomic radius; average ionic radius; all ionic radii of the most common ions; period, group, and block in the periodic table; molar volume; melting point; boiling point; polarizability; first ionization energy; and the number of valence s, p, and d electrons. Most of this information was collected using the python library PYMATGEN,[38] with the exception of the polarizability[69] and the ionization energies.[70] Of course, properties such as the number of valence electrons, the atomic size, or the electronegativity have been already used to model perovskites (see, e.g., the Goldschmidt's tolerance factor used to estimate the compatibility of an ion with the perovskite structure[71]) and, more generally, to understand and predict properties of materials (see, e.g., ref 72 and references therein).

5.1. Feature Importances. Besides choosing the optimal algorithm, the most important decision when using machine learning is the selection of the feature vector representing the problem. Considering the specific task of predicting the energy distance to the convex hull, the feature vector has to at least describe uniquely every compound. Our initial set of 119 features contains many closely related, and therefore highly correlated, properties (e.g., the atomic number and the atomic mass). It is therefore important to understand which ones are really relevant for the prediction of the stability of perovskites.

DOI: 10.1021/acs.chemmater.7b00156
Chem. Mater. 2017, 29, 5090−5103

To test the importance of the features, we used adaptive boosting with extremely randomized trees (boosted random forests were also tried with identical results). All the following MAEs are averaged over 20 training runs with randomly chosen training and test sets.

The starting selection of features resulted in a MAE of 130 meV/atom for the test and 1.7 meV/atom for the training set. It might be noted that one can already see the tendency of decision trees to overfitting even after using AdaBoost and extremely randomized trees to reduce the variance. In order to find the least influential features, the attribute *feature importances* of the regressor class in SCIKIT-LEARN was used. This attribute estimates the importance of the features by calculating out of bag errors. This is done by varying randomly one attribute for a random subset of the test set, while all other attributes are kept constant. Then the error of the inputs with the varied attribute is compared to the errors of the original inputs. The feature importances are the relative errors originating from this process.

We then removed the least important feature, retrained the regressor, and repeated the whole procedure. We found that we could eliminate most of the features without any noticeable deterioration of the predictive power. We found the lowest MAE of 121 meV/atom when using 11 features per element for a total of 33 features, namely (in no particular order), the following: (i) atomic number, (ii) Pauling electronegativity, (iii) most common oxidation state, (iv) average ionic radius, (v) number of valence electrons, (vi) period in the periodic table, (vii) group in the periodic table, (viii) ionization energy, (ix) polarizability, (x) number of s + p valence electrons, and (xi) number of d or f valence electrons. This is the set of features we used in the following. The number of input features can, however, be further decreased without a large increase in the MAE.

Surprisingly, the location of the elements in the periodic table is already sufficient to predict the energy distance to the convex hull with an MAE of 140 meV/atom. By adding the average ionic radius, the Pauling electronegativity, and the number of valence electrons, one can decrease the MAE to around 131 meV/atom. If one tries to use only the number of valence electrons and the period (MAE, 322 meV/atom) or the number of valence electrons and the group (MAE, 440 meV/atom), the MAE rises drastically. We believe this is due to the fact that the algorithm does not have enough information to completely distinguish all elements and therefore cannot identify all unique perovskites. The increase of the MAE when removing the period is larger than when removing the group, as the number of valence electrons and the group are closely related.

5.2. Hyperparameter Optimization. *Ridge Regression.* The model used by ridge regression is rather simple; therefore, it is not surprising to find that the error decreases monotonically with the number of features used for the fit. This is simply because every new feature is another coefficient for the linear regression enabling a more complicated model. The regularization parameter λ was also optimized, but it was more or less irrelevant as long as it was not zero. When λ was set equal to zero, the matrix was not invertible, showing that at least two features were linearly dependent. Furthermore, the MAE for the training set (304.1 ± 1.4 meV/atom) and the test set (306.0 ± 0.3 meV/atom) are almost the same, showing that overfitting was not a problem. However, the high MAE of the training set implies that this is a high-bias problem. This can be counteracted by using a more complex model, such as a higher order polynomial regression instead of a linear regression. In fact, when linear and squared feature terms are used, the MAE of the test set decreases to 298.9 ± 0.3 meV/atom. However, if one tries to include terms of higher polynomial order the regularization parameter λ has to be raised drastically to prevent the matrix from becoming noninvertible or the error from diverging. Unfortunately higher polynomial orders do not result in lower errors. Therefore, in the following, ridge regression with first and second order polynomial terms is used.

Random Forests and Extremely Randomized Trees. The most important hyperparameters that have to be fitted for random forests and extremely randomized trees are the number of trees in the ensemble and the percentage of features that are picked randomly to split a node. As expected, the error decreases with a larger number of trees but does not decrease significantly when using more than 350 trees. When using random forest regressors, it is usually recommended to use one-third of the features to determine the best split at each node.[73] We tested this hyperparameter and found that this was also the optimal split in our case. The final question regards the use of early stopping criteria. In order to solve this question, the MAE of the test set was calculated for training runs with different minimum numbers of samples per leaf. In all cases, early stopping criteria increased the testing error and were therefore not used.

Neural Networks. In this case, we used the same input data as those for the other machines, but the feature vectors were normalized, meaning that all values were transformed to the interval [−1, 1]. The energy distance to the convex hull was normalized to the range [−0.5:0.5]. We always normalized the training set and then applied the same normalization to the test set.

The final neural network that was used starts with a data layer which is a 33-dimensional feature vector (Figure 10). The

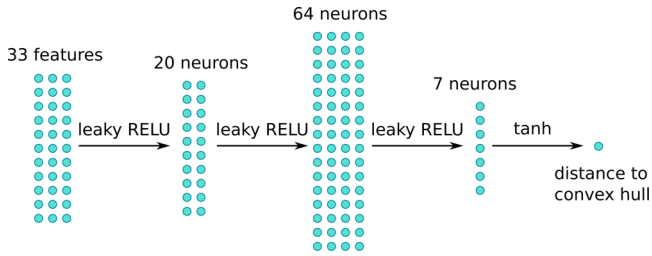


Figure 10. Architecture of the neural network used in this work.

data layer is connected to an inner-product layer which executes a matrix multiplication with a weight matrix $W_1 \in R^{20\times33}$. The resulting 20-dimensional vector was input into a leaky ReLU activation function. This process was repeated two more times ($W_2 \in R^{64\times20}$, $W_3 \in R^{7\times64}$). The fourth inner-product layer combines seven neurons into one value. A hyperbolic tangent is then used to calculate the energy distance to the convex hull from this value. We used an hyperbolic tangent because it is zero-centered and can, therefore, output both the positive and the negative values needed for the prediction of the energy distance to the convex hull. The neural network was trained using a squared error loss function and stochastic gradient descent.

In order to be able to train optimally the neural network, several hyperparameters had to be fitted. First, different learning rates and different numbers of epochs for each

DOI: 10.1021/acs.chemmater.7b00156
Chem. Mater. 2017, 29, 5090−5103

learning rate were tried out. The learning rate was gradually reduced from 0.1 to 0.00081. In Figure 11 the learning curve

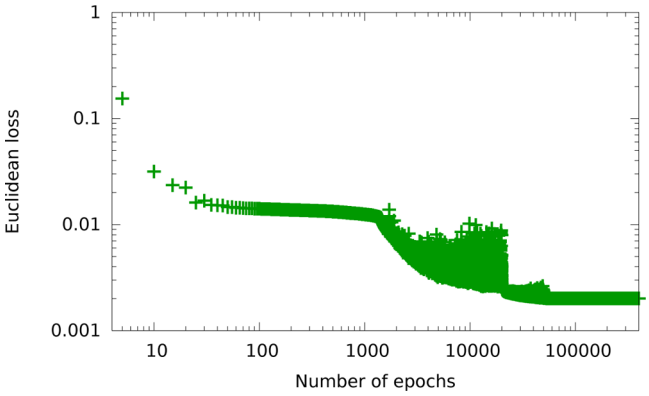


Figure 11. Euclidean loss plotted versus the number of epochs for a base learning rate of 0.1 which is gradually reduced by a factor of 0.3.

for the base learning rate 0.1 is plotted. This learning rate is reduced by a factor of 0.3 after a certain number of epochs. As one can see, for each learning rate the error converges to a new smaller level after a relatively small number of epochs and then oscillates around this level until a new learning rate is introduced. Once a new learning rate is introduced, the error decreases drastically. We see convergence at every learning rate because once the error is small enough the weight changes Δw_{ij} are large, and the gradient constantly changes direction. Our neural networks were trained using this procedure.

The second hyperparameter is the regularization parameter. We trained neural networks with 15 different values for λ:

$$\lambda = 0.0005 \times 2^i, \qquad i \in \{0, 1, ..., 14\} \tag{15}$$

We found that the minimum error for our problem was achieved for $\lambda = 0.004$. Finally, the best value of the momentum in the stochastic gradient descent used to train the networks turned out to be 0.95.

Adaptive Boosting. AdaBoost was used with random forests and extremely randomized trees. In both cases, we tested whether the hyperparameters should be changed in comparison to the standalone algorithms without AdaBoost. The number of trees was reduced to 250 as the MAE converges faster when using AdaBoost. Of course, the actual number of trees is still larger because each boosted regressor contains multiple random forests/extremely randomized trees ensembles. The percentage of features used to find a split was left unaltered as the results for AdaBoost are qualitatively equal to the ones of the standalone algorithms. The same holds true for early stopping criteria.

5.3. Comparison of the Methods. Table 1 shows the MAEs and their standard deviations for all machine learning models averaged over 20 training and testing runs with random training and testing sets. The hyperparameters of the algorithms were all set according to the results of section 5.2. As expected, ridge regression produces by far the worst predictions, as its underlying model is just not complex enough to fit E_{hull}. Neural networks yield a considerably better result, almost as good as decision tree algorithms. It is clear that neural networks are capable of the most complex models; however, it is also known that it is rather hard (and time-consuming) to find the optimal network configuration and to train it. It is also very likely that neural networks, such as the handful of different

Table 1. MAEs and Standard Deviations (meV/atom) for All Tested Machine Learning Models Averaged over 20 Training and Testing Runs, Sorted from Largest to Smallest Error

MAE	machine learning model
298.9 ± 0.3	ridge regression
155.5 ± 4.8	neural network
140.0 ± 0.6	random forests
126.6 ± 1.0	AdaBoost/random forests
123.1 ± 0.8	extremely randomized trees
121.3 ± 0.8	AdaBoost/extremely randomized trees

topologies that we experimented with, would benefit from considerably larger training sets. It is therefore not surprising that these methods failed to perform as well as the much more straightforward decision trees.

When comparing the decision tree algorithms, we see that extremely randomized tree ensembles perform consistently better than random forests. The former method is favored due to the high-variance nature of the problem, as randomized tree ensembles are better in reducing this quantity[59] than random forests. Introducing adaptive boosting decreases the MAE of the random forests by around 10%, while not bringing any improvements for extremely randomized trees. This implies that further variance reduction measures will most likely not bring any more improvements for extremely randomized trees.

We emphasize that we are trying to reproduce DFT energies and not experimental ones (due to the lack of available experimental data as discussed before). However, if we assume that they both follow essentially the same statistical distribution, we can try to extract a few comparisons between the methods. The usually cited error for the cohesive energies calculated with the Perdew−Burke−Ernzerhof approximation is around 250 meV/atom.[74,75] This error can be reduced by a factor of 2−3 by using other functionals which are more adapted to the calculation of cohesive energies.[74,76] There are few works in the literature that deal with the accuracy of density-functional theory in calculating formation energies and therefore the energy distance to the convex hull of stability,[77] but we can safely assume that the error can in this case decrease by a certain margin. Comparing with the error of ≈120 meV that we found in this work, we can safely conclude that extremely randomized trees yield an average error that is perfectly in line with the best DFT approaches for a small fraction of the computational effort. Moreover, this error can be easily decreased simply by increasing the size of the training set, as we will see in the following section.

5.4. Analysis of the Errors. Besides the choice of the algorithm, machine learning predictions depend dramatically on the size and the quality of the training set. It is obviously important to verify how the error in the prediction of E_{hull} evolves with the size of the training set. In order to obtain this information, we trained extremely random trees with the increasing size of the training set (from 1000 to 50000 samples). The machines were then tested on another 180000 samples. The results can be found in Figure 12. Obviously, the MAE decreases monotonically with the size of the training set, following a curve that decays with a power of the size of the training set, namely, MAE = $a \times \text{size}^{-b}$ with $a = 2286 \pm 40$ meV/atom and $b = 0.297 \pm 0.002$. This means that doubling the training set only decreases the error by around 20%.

DOI: 10.1021/acs.chemmater.7b00156
Chem. Mater. 2017, 29, 5090−5103

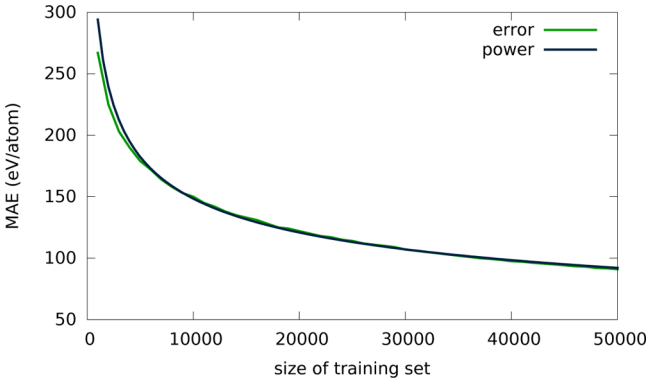


Figure 12. MAE (meV/atom) of the test set for AdaBoost used with extremely random tress plotted against the size of the training set.

To compare the different methods, we used what we can call a moderate size of 20000 samples. From Figure 12 we can infer that this number is a rather good compromise between the size of the training data and the computer time required to generate it. We note, however, that this is true for (i) a ternary compound with (ii) a fixed (perovskite) structure. We can obviously expect that the size of the training set required to obtain a certain MAE will increase with the number of different elements in the unit cell and the structural variety. Much more data are, however, required in order to verify how rapid this increase would be.

We can take our analysis a step further and study how each individual element of the periodic table contributes to the error. It is well-known that there are regions of the periodic table where the properties of the elements vary in a rather smooth and predictable way, while there are other elements that behave in more unique manners. To display this information, we averaged the error in the prediction of E_{hull} for all systems containing a certain element. The results are depicted in Figure 13. We note that these results are qualitatively similar for random forests with AdaBoost.

From Figure 13 it is clear that the compounds containing elements of the first row of the periodic table have a rather large error that increases with increasing electronegativity. We believe that this error is related to the so-called "first-row anomaly": the properties of Li through Ne are significantly different from the other elements in the respective groups,[78] and are therefore more difficult to predict. Perhaps counter-intuitively, it is rather the heavy elements which behave in an easily predictive way, and not the more familiar elements of the first row.[79]

Also chromium and manganese have a MAE which is significantly higher than the average MAE. This is probably related to the fact that these elements often form complicated magnetic structures, whose subtleties were difficult to capture by our machines. Finally we find that cesium also exhibits a rather large error. In principle, from the chemical point of view there is nothing strange with cesium, being only a larger version of rubidium and potassium. To find the source of the considerable error, one has to look, in our opinion, to the numerical setup used in the high-throughput search. As we mentioned before, the pseudopotential for cesium in version 5.2 of VASP has several problems, often leading to crashes and difficult convergence. We are therefore convinced that this error is simply due to the low quality of the pseudopotential.

6. CONCLUSIONS

In this Article we performed an extensive benchmark of machine learning methods for the prediction of energetics of inorganic solid-state phases. To have a fully unbiased assessment, and to circumvent the lack of extensive experimental data, we started by developing a synthetic benchmark. This consisted of all possible compounds with ABX_3 stoichiometry in the cubic perovskite structure. We took into account a large portion of the periodic table (64 elements), amounting to almost 250000 systems including both standard and inverted perovskites. The equilibrium geometry and energy of these solid phases were then evaluated using density-functional theory. From the total energy we then obtained the cohesive energy and the energy distance to the convex hull of stability. We believe that the existence of such a large, consistent data set is already an interesting development in itself as it allowed us (i) to perform an interesting statistical analysis of the data and (ii) to perform a direct comparison of regression methods in a controlled environment. As such, and in order to stimulate research in these methods, the full data set will be made freely available.[80]

Then we used these data to benchmark different machine learning methods, namely, ridge regression, neural networks, random forest, and extremely randomized trees. The most accurate method turns out to be extremely randomized trees, followed by random forests. Ridge regression yields rather large errors, mainly because it relies on a too simple model for our problem. Finally, neural networks are very hard to optimize and

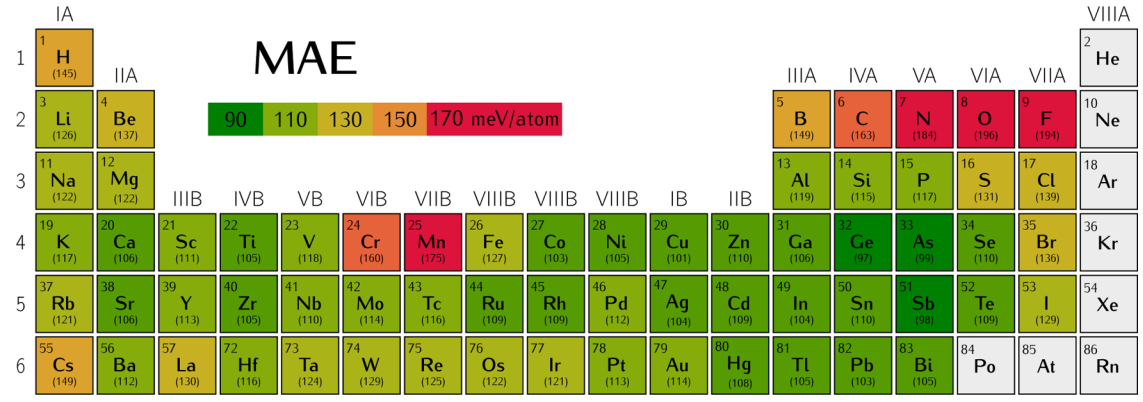


Figure 13. MAE (meV/atom) of the test set for AdaBoost used with extremely random tress averaged over all compounds containing each element of the periodic table. The training set had 20000 samples. The numbers in parentheses are the actual MAE for each element (see text for details).

DOI: 10.1021/acs.chemmater.7b00156
Chem. Mater. 2017, 29, 5090−5103

to train and would probably require considerably more test data to fully unleash their potential. We also found that the use of adaptive boosting helps to improve the random forest prediction, but it only decreases marginally the error of extremely randomized trees.

Another interesting aspect regards the feature vector necessary to perform the prediction of E_{hull}. Tens, if not hundreds, of properties of the elements and of their elementary substances are readily available to be used as input for the machines. However, our calculations indicate that two numbers, namely, the period and the group of the element in the periodic table, are by far the most important features (yielding an error of ≈140 meV/atom for a training set of 20000 samples). Adding nine extra properties, we can reach an error of 121 meV/atom. In some sense, the machine seems to be able to reconstruct most other properties from the simple position of the elements in the periodic table, rendering their explicit inclusion redundant.

Finally, we studied how the error decreases with the size of the training set. To reach an average precision of 130−100 meV/atom, one requires around 20000−30000 training samples. Unfortunately, the error decreases slowly for larger sets. Furthermore, this error is considerably larger if the compound includes elements from the first row (very likely due to the so-called first-row anomaly in chemistry) or a few transition metals (that often yield complicated magnetic structures).

From these numbers we can propose how machine learning methods can be reliably used to accelerate DFT-based high-throughput studies. As an example, we again use the case of perovskites. One could (i) perform a DFT calculation of around 20000 cubic perovskites with random compositions, (ii) train a regressor using extremely random trees, (iii) use this regressor to predict the energy distance to the hull of the remaining ≈230000 possible compounds without performing on them explicit DFT calculations, (iv) remove all systems that lie higher than a certain cutoff energy form the hull (for example, putting the cutoff at 700 meV/atom allows us to recover 99.4% of all systems that are stable within DFT), and (v) calculate, within DFT, these compounds. In our example, this amounts to an extra 41000 DFT calculations, leading to a total saving of computation time of 73%. We can further reduce the computational effort by allowing for a larger number of false negatives. In fact, to recover 99% of all stable systems, we can save 78% of the computational effort, while to recover 95%, we save 86%.

In view of these encouraging results and thanks to the understanding gained through their analysis, we are confident that machine learning techniques will have a bright future in computational materials science and, in particular, in ab initio high-throughput initiatives.

AUTHOR INFORMATION

Corresponding Author

*E-mail: miguel.marques@physik.uni-halle.de.

ORCID

Silvana Botti: 0000-0002-4920-2370

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

J.S. acknowledges financial support from the China Scholarship Council. M.A.L.M acknowledges partial support from the DFG though Projects SFB-762 and MA-6786/1. Computational resources were provide by the Leibniz Supercomputing Centre through the SuperMuk Projects p1841a and pr48je.

REFERENCES

(1) Marsland, S. *Machine learning: An algorithmic perspective*; CRC Press, 2015.

(2) Sun, Y.; Wang, X.; Tang, X. Deep learning face representation from predicting 10,000 classes. *IEEE conference on computer vision and pattern recognition (CVPR)*; IEEE, 2014; pp 1891−1898, DOI: 10.1109/CVPR.2014.244.

(3) Schroff, F.; Kalenichenko, D.; Philbin, J. FaceNet: A unified embedding for face recognition and clustering. *IEEE conference on computer vision and pattern recognition (CVPR)*; IEEE, 2015; pp 815−823, DOI: 10.1109/CVPR.2015.7298682.

(4) Weyand, T.; Kostrikov, I.; Philbin, J. PlaNet-photo geolocation with convolutional neural networks. 2016, arXiv:1602.05314; DOI: 10.1007/978-3-319-46484-8_3.

(5) Bojarski, M.; Del Testa, D.; Dworakowski, D.; Firner, B.; Flepp, B.; Goyal, P.; Jackel, L. D.; Monfort, M.; Muller, U.; Zhang, J.; Zhang, X.; Zhao, J.; Zieba, K. End to end learning for self-driving cars. 2016, arXiv:1604.07316.

(6) Silver, D.; et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **2016**, *529*, 484−489.

(7) 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.* **2012**, *108*, 058301.

(8) Huan, T. D.; Mannodi-Kanakkithodi, A.; Ramprasad, R. Accelerated materials property predictions and design using motif-based fingerprints. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2015**, *92*, 014106.

(9) Mannodi-Kanakkithodi, A.; Pilania, G.; Huan, T. D.; Lookman, T.; Ramprasad, R. Machine learning strategy for accelerated design of polymer dielectrics. *Sci. Rep.* **2016**, *6*, 20952.

(10) Pilania, G.; Wang, C.; Jiang, X.; Rajasekaran, S.; Ramprasad, R. Accelerating materials property predictions using machine learning. *Sci. Rep.* **2013**, *3*, 2810.

(11) Pozun, Z. D.; Hansen, K.; Sheppard, D.; Rupp, M.; Müller, K.-R.; Henkelman, G. Optimizing transition states via kernel-based machine learning. *J. Chem. Phys.* **2012**, *136*, 174101.

(12) Behler, J. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *J. Chem. Phys.* **2011**, *134*, 074106.

(13) Botu, V.; Ramprasad, R. Adaptive machine learning framework to accelerate ab initio molecular dynamics. *Int. J. Quantum Chem.* **2015**, *115*, 1074−1083.

(14) Pilania, G.; Mannodi-Kanakkithodi, A.; Uberuaga, B.; Ramprasad, R.; Gubernatis, J.; Lookman, T. Machine learning bandgaps of double perovskites. *Sci. Rep.* **2016**, *6*, 19375.

(15) Lee, J.; Seko, A.; Shitara, K.; Tanaka, I. Prediction model of band-gap for AX binary compounds by combination of density functional theory calculations and machine learning techniques. 2015, arXiv:1509.00973; DOI: 10.1103/PhysRevB.93.115104.

(16) Dey, P.; Bible, J.; Datta, S.; Broderick, S.; Jasinski, J.; Sunkara, M.; Menon, M.; Rajan, K. Informatics-aided bandgap engineering for solar materials. *Comput. Mater. Sci.* **2014**, *83*, 185−195.

(17) Faber, F. A.; Lindmaa, A.; von Lilienfeld, O. A.; Armiento, R. Machine learning energies of 2 million elpasolite (ABC_2D_6) crystals. *Phys. Rev. Lett.* **2016**, *117*, 135502.

(18) Meredig, B.; Agrawal, A.; Kirklin, S.; Saal, J. E.; Doak, J. W.; Thompson, A.; Zhang, K.; Choudhary, A.; Wolverton, C. Combinatorial screening for new materials in unconstrained composition space with machine learning. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2014**, *89*, 094104.

DOI: 10.1021/acs.chemmater.7b00156
Chem. Mater. 2017, 29, 5090−5103

(19) Oliynyk, A. O.; Antono, E.; Sparks, T. D.; Ghadbeigi, L.; Gaultois, M. W.; Meredig, B.; Mar, A. High-throughput machine-learning-driven synthesis of full-Heusler compounds. *Chem. Mater.* **2016**, *28*, 7324−7331.
(20) Ghiringhelli, L. M.; Vybiral, J.; Levchenko, S. V.; Draxl, C.; Scheffler, M. Big data of materials science: Critical role of the descriptor. *Phys. Rev. Lett.* **2015**, *114*, 105503.
(21) Faber, F.; Lindmaa, A.; von Lilienfeld, O. A.; Armiento, R. Crystal structure representations for machine learning models of formation energies. *Int. J. Quantum Chem.* **2015**, *115*, 1094−1101.
(22) Schütt, K. T.; Glawe, H.; Brockherde, F.; Sanna, A.; Müller, K. R.; Gross, E. K. U. How to represent crystal structures for machine learning: Towards fast prediction of electronic properties. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2014**, *89*, 205118.
(23) Morgan, D.; Ceder, G.; Curtarolo, S. High-throughput and data mining with ab initio methods. *Meas. Sci. Technol.* **2005**, *16*, 296.
(24) Ceder, G. Opportunities and challenges for first-principles materials design and applications to Li battery materials. *MRS Bull.* **2010**, *35*, 693−701.
(25) Zhang, X.; Wang, Y.; Lv, J.; Zhu, C.; Li, Q.; Zhang, M.; Li, Q.; Ma, Y. First-principles structural design of superhard materials. *J. Chem. Phys.* **2013**, *138*, 114101.
(26) Katayama-Yoshida, H.; Sato, K.; Kizaki, H.; Funashima, H.; Hamada, I.; Fukushima, T.; Dinh, V.; Toyoda, M. Ab initio materials design for transparent-conducting-oxide-based new-functional materials. *Appl. Phys. A: Mater. Sci. Process.* **2007**, *89*, 19−27.
(27) Hautier, G.; Miglio, A.; Ceder, G.; Rignanese, G.-M.; Gonze, X. Identification and design principles of low hole effective mass p-type transparent conducting oxides. *Nat. Commun.* **2013**, *4*, 2292.
(28) Sarmiento-Perez, R.; Cerqueira, T. F.; Körbel, S.; Botti, S.; Marques, M. A. Prediction of stable nitride perovskites. *Chem. Mater.* **2015**, *27*, 5957−5963.
(29) Sharma, V.; Wang, C.; Lorenzini, R. G.; Ma, R.; Zhu, Q.; Sinkovits, D. W.; Pilania, G.; Oganov, A. R.; Kumar, S.; Sotzing, G. A.; Boggs, S. A.; Ramprasad, R. Rational design of all organic polymer dielectrics. *Nat. Commun.* **2014**, *5*, 4845.
(30) Drebov, N.; Martinez-Limia, A.; Kunz, L.; Gola, A.; Shigematsu, T.; Eckl, T.; Gumbsch, P.; Elsässer, C. Ab initio screening methodology applied to the search for new permanent magnetic materials. *New J. Phys.* **2013**, *15*, 125023.
(31) Kresse, G.; Furthmüller, J. Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set. *Comput. Mater. Sci.* **1996**, *6*, 15−50.
(32) Kresse, G.; Furthmüller, J. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1996**, *54*, 11169−11186.
(33) Jain, A.; Ong, S. P.; Hautier, G.; Chen, W.; Richards, W. D.; Dacek, S.; Cholia, S.; Gunter, D.; Skinner, D.; Ceder, G.; Persson, K. A. The Materials Project: A materials genome approach to accelerating materials innovation. *APL Mater.* **2013**, *1*, 011002.
(34) 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* **2013**, *65*, 1501−1509.
(35) Perdew, J. P.; Burke, K.; Ernzerhof, M. Generalized gradient approximation made simple. *Phys. Rev. Lett.* **1996**, *77*, 3865−3868.
(36) Blöchl, P. E. Projector augmented-wave method. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1994**, *50*, 17953.
(37) Curtarolo, S.; Setyawan, W.; Wang, S.; Xue, J.; Yang, K.; Taylor, R. H.; Nelson, L. J.; Hart, G. L.; Sanvito, S.; Buongiorno-Nardelli, M.; Mingo, N.; Levy, O. AFLOWLIB.ORG: A distributed materials properties repository from high-throughput ab initio calculations. *Comput. Mater. Sci.* **2012**, *58*, 227−235.
(38) Ong, S. P.; Richards, W. D.; Jain, A.; Hautier, G.; Kocher, M.; Cholia, S.; Gunter, D.; Chevrier, V. L.; Persson, K. A.; Ceder, G. Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis. *Comput. Mater. Sci.* **2013**, *68*, 314−319.
(39) Bergerhoff, G.; Brown, I. In *Crystallographic Databases*; Allen, F., Bergerhoff, G., Sievers, R., Eds.; International Union of Crystallography: Chester, U.K., 1987.
(40) Belsky, A.; Hellenbrandt, M.; Karen, V. L.; Luks ch, P. New developments in the Inorganic Crystal Structure Database (ICSD): accessibility in support of materials research and design. *Acta Crystallogr., Sect. B: Struct. Sci.* **2002**, *58*, 364−369.
(41) Schaak, R.; Avdeev, M.; Lee, W.-L.; Lawes, G.; Zandbergen, H.; Jorgensen, J.; Ong, N.; Ramirez, A.; Cava, R. Formation of transition metal boride and carbide perovskites related to superconducting $MgCNi_3$. *J. Solid State Chem.* **2004**, *177*, 1244−1251.
(42) He, T.; Huang, Q.; Ramirez, A.; Wang, Y.; Regan, K.; Rogado, N.; Hayward, M.; Haas, M.; Slusky, J.; Inumara, K.; Zandbergen, H. W.; Ong, N. P.; Cava, R. J. Superconductivity in the non-oxide perovskite $MgCNi_3$. *Nature* **2001**, *411*, 54−56.
(43) Mao, Z. Q.; Rosario, M. M.; Nelson, K. D.; Wu, K.; Deac, I. G.; Schiffer, P.; Liu, Y.; He, T.; Regan, K. A.; Cava, R. J. Experimental determination of superconducting parameters for the intermetallic perovskite superconductor $MgCNi_3$. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2003**, *67*, 094502.
(44) García, J.; Bartolomé, J.; Goñzález, D.; Navarro, R.; Fruchart, D. Thermophysical properties of intermetallic Mn_3MC perovskites I. Heat capacity of manganese gallium carbide Mn_3GaC. *J. Chem. Thermodyn.* **1983**, *15*, 1059−1069.
(45) Kaneko, T.; Kanomata, T.; Shirakawa, K. Pressure Effect on the Magnetic Transition Temperatures in the Intermetallic Compounds Mn_3MC (M = Ga, Zn and Sn). *J. Phys. Soc. Jpn.* **1987**, *56*, 4047−4055.
(46) Connétable, D.; Maugis, P. First principle calculations of the κ-Fe_3AlC perovskite and iron-aluminium intermetallics. *Intermetallics* **2008**, *16*, 345−352.
(47) Bartell, L. S. Molecular geometry: Bonded versus nonbonded interactions. *J. Chem. Educ.* **1968**, *45*, 754.
(48) Pearson, R. G. The principle of maximum hardness. *Acc. Chem. Res.* **1993**, *26*, 250−255.
(49) Burdett, J. K.; Coddens, B. A.; Kulkarni, G. V. Band gap and stability of solids. *Inorg. Chem.* **1988**, *27*, 3259−3261.
(50) Carrete, J.; Li, W.; Mingo, N.; Wang, S.; Curtarolo, S. Finding unprecedentedly low-thermal-conductivity half-Heusler semiconductors via high-throughput materials modeling. *Phys. Rev. X* **2014**, *4*, 011019.
(51) *NOMAD Repository*, http://nomad-repository.eu/.
(52) Ng, A. Y. Feature selection, L_1 vs. L_2 regularization, and rotational invariance. *Proceedings of the twenty-first international conference on machine learning*; ACM, 2004; p 78, DOI: 10.1145/1015330.1015435.
(53) Quinlan, J. R. Simplifying decision trees. *Int. J. Man-Mach. Stud.* **1987**, *27*, 221−234.
(54) Breiman, L.; Friedman, J.; Stone, C. J.; Olshen, R. A. *Classification and regression trees*; CRC Press, 1984.
(55) Quinlan, J. R. Induction of decision trees. *Mach. Learn.* **1986**, *1*, 81−106.
(56) Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5−32.
(57) Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123−140.
(58) Timofeev, R. Classification and regression trees (cart) theory and applications. Master's thesis, Humboldt University, Berlin, Germany, 2004.
(59) Geurts, P.; Ernst, D.; Wehenkel, L. Extremely randomized trees. *Mach. Learn.* **2006**, *63*, 3−42.
(60) Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157−166.
(61) LeCun, Y.; Kanter, I.; Solla, S. Second-order properties of error surfaces: learning time and generalization. *Advances in neural information processing systems 3*, Denver, CO, USA; Neural Information Processing Systems Foundation, 1990; pp 918−924.
(62) Krizhevsky, A.; Sutskever, I.; Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in*

DOI: 10.1021/acs.chemmater.7b00156
Chem. Mater. 2017, 29, 5090−5103

neural information processing systems 25; Neural Information Processing Systems Foundation, 2012; pp 1097−1105.
(63) Maas, A. L.; Hannun, A. Y.; Ng, A. Y. Rectifier nonlinearities improve neural network acoustic models. *Proceedings of the 30th international conference on machine learning*; Association for Computational Linguistics, 2013.
(64) He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. *IEEE international conference on computer vision (ICCV)*; IEEE Computer Society, 2015; pp 1026−1034, DOI: 10.1109/ICCV.2015.123.
(65) Drucker, H. Improving regressors using boosting techniques. *Proceedings of the fourteeth international conference on machine learning*; Morgan Kaufmann, 1997; pp 107−115.
(66) Freund, Y.; Schapire, R. E. Experiments with a new boosting algorithm. *Proceedings of the thirteenth international conference on machine learning*; Saitta, L., Ed.; Morgan Kaufmann, 1996; pp 148−156.
(67) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825−2830.
(68) Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional architecture for fast feature embedding. 2014, arXiv:1408.5093.
(69) Schwerdtfeger, P. Table of experimental and calculated static dipole polarizabilities for the electronic ground states of the neutral elements (in atomic units). http://ctcp.massey.ac.nz/index.php?menu/dipole&page/dipole, 2014; Accessed: Feb. 8, 2014.
(70) Kramida, A.; Ralchenko, Yu.; Reader, J.; NIST ASD Team. *NIST Atomic Spectra Database* (ver. 5.3), [Online];National Institute of Standards and Technology, Gaithersburg, MD, USA, 2015; Available: http://physics.nist.gov/asd [Sep. 14, 2016].
(71) Goldschmidt, V. ie Gesetze der Krystallochemie. *Naturwissenschaften* **1926**, *14*, 477−485.
(72) Pettifor, D. A chemical scale for crystal-structure maps. *Solid State Commun.* **1984**, *51*, 31−34.
(73) Liaw, A.; Wiener, M. Classification and regression by randomForest. *R News* **2013**, *2*, 18−22.
(74) Sarmiento-Pérez, R.; Botti, S.; Marques, M. A. L. Optimized exchange and correlation semilocal functional for the calculation of energies of formation. *J. Chem. Theory Comput.* **2015**, *11*, 3844−3850.
(75) Stevanović, V.; Lany, S.; Zhang, X.; Zunger, A. Correcting density functional theory for accurate predictions of compound enthalpies of formation: Fitted elemental-phase reference energies. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2012**, *85*, 115104.
(76) Tran, F.; Blaha, P.; Betzinger, M.; Blügel, S. Comparison between exact and semilocal exchange potentials: An all-electron study for solids. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2015**, *91*, 165121.
(77) Hautier, G.; Ong, S. P.; Jain, A.; Moore, C. J.; Ceder, G. Accuracy of density functional theory in predicting formation energies of ternary oxides from binary oxides and its implication on phase stability. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2012**, *85*, 155208.
(78) Miessler, G. L.; Fischer, P. J.; Tarr, D. A. *Inorganic Chemistry: Pearson New International Edition*; Pearson Higher Education, 2013.
(79) Kutzelnigg, W. Chemical Bonding in Higher Main Group Elements. *Angew. Chem., Int. Ed. Engl.* **1984**, *23*, 272−295.
(80) Schmidt, J.; Shi, J.; Borlido, P.; Chen, L.; Botti, S.; Marques, M. Dataset of 250000 cubic perovskites. http://tddft.org/bmg/, 2017. The full dataset will be available on our web page.

DOI: 10.1021/acs.chemmater.7b00156
Chem. Mater. 2017, 29, 5090−5103