International Journal of QUANTUM CHEMISTRY

Crystal Structure Representations for Machine Learning Models of Formation Energies

Felix Faber,[a] Alexander Lindmaa,[b] O. Anatole von Lilienfeld,*[a,c] and Rickard Armiento*[b]

We introduce and evaluate a set of feature vector representations of crystal structures for machine learning (ML) models of formation energies of solids. ML models of atomization energies of organic molecules have been successful using a Coulomb matrix representation of the molecule. We consider three ways to generalize such representations to periodic systems: (i) a matrix where each element is related to the Ewald sum of the electrostatic interaction between two different atoms in the unit cell repeated over the lattice; (ii) an extended Coulomb-like matrix that takes into account a number of neighboring unit cells; and (iii) an *ansatz* that mimics the periodicity and the basic features of the elements in the Ewald sum matrix using a sine function of the crystal coordinates of the atoms. The representations are compared for a Laplacian kernel with Manhattan norm, trained to reproduce formation energies using a dataset of 3938 crystal structures obtained from the Materials Project. For training sets consisting of 3000 crystals, the generalization error in predicting formation energies of new structures corresponds to (i) 0.49, (ii) 0.64, and (iii) 0.37 eV/atom for the respective representations.

DOI: 10.1002/qua.24917

Introduction

First-principles simulations for the prediction of properties of chemical and materials systems have become a standard tool throughout theoretical chemistry, physics, and biology. These simulations are based on repeatedly solving numerical approximations to the underlying many-body problem, that is, the Schrödinger equation. However, in the pursuit of ever increasing efficiency, there is a growing interest in sidestepping the physics-based formulation of the problem, and instead exploit big data methodology, for example, artificial intelligence, evolutionary, and machine learning (ML) schemes. If successful, such approaches can lead to an orders-of-magnitude increase in computational efficiency for describing properties of atomistic systems. Such a change would not merely bring incremental progress, but certainly represents a paradigm-shift in terms of enabling the study of systems and problems which hitherto have been completely out of reach.

Recently, one of us has presented an ML scheme that predicts atomization energies of new (out-of-sample) molecules with an accuracy even beyond that of the most common first-principles method, density functional theory (DFT).[1–4] Subsequently, it was shown that this approach also works for other properties such as molecular polarizability and frontier orbital eigenvalues,[5] transmission coefficients of electron transport in doped nano-ribbon models,[6] and even densities of state within the Andersion impurity model.[7] A critical component of any such ML scheme is the feature vector representation of the dataset, also often called the *descriptor*;[8,9] that is, the mapping of the atomistic description of the system into a form suitable for matrix operations. In the cited work,[1] an atom by atom matrix, dubbed *Coulomb matrix*, was used with diagonal and off-diagonal elements representing the potential of the free atom and the interatomic Coulomb repulsion between nuclear charges.

The aim of this study is to consider various ways of adapting this representation to periodic systems, and to compare their performance in ML models of formation energies of solids. We have considered three generalizations of the Coulomb matrix idea; (i) a matrix where each element is related to the Ewald sum of the electrostatic interaction between two different atoms in the unit cell repeated over the lattice; (ii) an extended Coulomb-like matrix that takes into account a number of neighboring unit cells; and (iii) a simplified matrix

[a] *F. Faber, O. Anatole von Lilienfeld*
Department of Chemistry and Institute of Physical Chemistry and National Center for Computational Design and Discovery of Novel Materials, University of Basel, Switzerland*
[b] *A. Lindmaa and R. Armiento*
Department of Physics, Chemistry and Biology, Linköping University, SE-581 83, Linköping, Sweden
E-mail: rickard.armiento@liu.se
[c] *O. Anatole von Lilienfeld*
Argonne Leadership Computing Facility and Argonne National Laboratory, 9700 S. Cass Avenue, Lemont, Illinois, 60439*
E-mail: anatole.vonlilienfeld@unibas.ch

Contract grant sponsor: Swedish Research Council (VR); contract grant number: 621-2011-4249 (R.A.).
Contract grant sponsor: Linnaeus Environment at Linköping on Nanoscale Functional Materials (Funded by VR).
Contract grant sponsor: Swiss National Science foundation; contract grant number: PP00P2_138932 (O.A.v.L.).
Contract grant sponsor: Office of Science of the U.S. DOE; contract grant number: DE-AC02-06CH11357.
Contract grant sponsor: Air Force Office of Scientific Research, Air Force Material Command, USAF; contract grant number: FA9550-15-1-0026.

ansatz chosen to mimic the periodicity and basic features of the elements in the Ewald sum matrix using a sine function of atomic coordinates.

A few studies have already considered representations of periodic systems suitable for ML. Schutt et al. trained an ML model using radial distribution functions to predict the density of states at the Fermi level of solids.[4] Meredig et al. used a heuristics-based ML model to screen 1.6 M ternary solids for stability.[10] Ramprasad and coworkers, relying on chemo-structural fingerprints in ML models of formation energies (and other properties) of polymers, obtained scatter plots[11] for which visual inspection suggests an error on the order of magnitude of ~0.5 eV. Bartok et al. used ML to enhance the computer simulation of molecular materials.[12] However, to the best of our knowledge, no clear example has been presented so far for directly applying ML to reproduce cell-, cohesive-, or formation energies based on an atom by atom matrix-based representation of the crystal structure. We present such an application based on the ML methods which have shown good performance for molecules. We find that all representations investigated in this work yield similar accuracy (0.4−0.6 eV/atom at 3000 crystals), which is less impressive than the accuracy found for atomization energies of molecules (0.02 eV/atom for training sets of similar size). However, our results indicate that if the dataset used for training can be further expanded, a machine capable of estimating formation energies at near first-principles accuracy is still within reach. We briefly discuss the reason for the disparity in performance between finite and periodic systems, and how this may be resolved.

This article is organized as follows. In kernel ridge regression section, we briefly present the kernel ridge regression (KRR) scheme. In representations section, we introduce our various representations of crystal structures. In implementation details and datasets section, some technical details are given regarding the implementation. In results section, we analyze the representations and test their performance in machines trained on a dataset of formation energies of solids. In discussion section, we discuss our results, and finally, in summary and conclusions section, this study is summarized and concluded.

Kernel Ridge Regression

We present a short summary of the ridge regression method[13] and KRR,[14,15] defining our notation. In ridge regression, one seeks to approximate an unknown function $y(x)$, where x is a feature-vector representation of some input (e.g., a molecule or crystal system). We start from a set of n data points $\mathbf{y}=(y_1, y_2, ..., y_n)^{\mathrm{T}}$ known for a set of feature vectors $\mathbf{x}=(x_1, x_2, ..., x_n)^{\mathrm{T}}$, where we use the linear-algebra convention of distinguishing row and column vectors, and a^{T} is the transpose of a.

An approximation $f(x,\boldsymbol{\alpha})$ is constructed as

$$f(x,\boldsymbol{\alpha})=\mathbf{k}(x)\boldsymbol{\alpha}=\sum_{j=1}^{n}k_j(x)\alpha_j \tag{1}$$

$$\boldsymbol{\alpha}=(\alpha_1, \alpha_2, ..., \alpha_n) \tag{2}$$

$$\mathbf{k}(x)=(k_1(x), k_2(x), ..., k_n(x))^{\mathrm{T}}, \tag{3}$$

where $k_j(x)$ is a function *ansatz* of x which is the same for all j, and to be chosen freely, usually assumed to be polynomial. The approximation $f(x,\boldsymbol{\alpha})$ is optimized by seeking the $\boldsymbol{\alpha}$ that minimizes the Euclidean norm $\|\mathbf{y}-\mathbf{k}(\mathbf{x})\boldsymbol{\alpha}\|_2$. The solution is given by the normal equation

$$\boldsymbol{\alpha}=(\mathbf{k}(\mathbf{x})^{\mathrm{T}}\mathbf{k}(\mathbf{x}))^{-1}\mathbf{k}(\mathbf{x})^{\mathrm{T}}\mathbf{y}. \tag{4}$$

If an *ansatz* for $\mathbf{k}(x)$ with a high degree of freedom is used, there is considerable risk of overfitting, that is, one arrives at a model that fits the training dataset well, but still performs poorly when predicting y for out-of-sample cases (feature vectors x that are not included in the training set $\mathbf{x}$.) Several techniques are used to address this problem, for example, cross-validation and regularization. In the case of the latter, Eq. (4) is modified by inserting an extra term,

$$\boldsymbol{\alpha}=(\mathbf{k}(\mathbf{x})^{\mathrm{T}}\mathbf{k}(\mathbf{x})+\lambda I)^{-1}\mathbf{k}(\mathbf{x})^{\mathrm{T}}\mathbf{y}, \tag{5}$$

where λ is the regularization parameter and I the identity matrix.

Ridge regression is extended into KRR by introducing a map, Φ^K, from a nonlinear space to a linear space that is known as the feature space.[16–18] The mapping Φ^K is usually nontrivial, but does not need to be known explicitly as it is sufficient to know the inner product between the mapped data, $\langle\Phi^K(x_i),\Phi^K(x_j)\rangle=\sum\Phi^K(x_i)_l\Phi^K(x_j)_l=k(x_i,x_j)$where $k(x_i,x_j)$ is the kernel. This procedure is also known as the *kernel trick*.[19] The function $f(x,\boldsymbol{\alpha})$ now takes the form

$$f(x,\boldsymbol{\alpha})=\sum_{i}^{n}\alpha_i k(x,x_i) \tag{6}$$

and the approximation is optimized by seeking the minimum of

$$\sum_{i}^{n}(y_i-f(x_i,\boldsymbol{\alpha}))^2+\sum_{i,j}^{n}\alpha_i k(x_j,x_i)\alpha_j, \tag{7}$$

with solution

$$\boldsymbol{\alpha}=(\mathbf{K}+\lambda I)^{-1}\mathbf{y}, \tag{8}$$

where $\mathbf{K}=k(x_i,x_j)$ is the kernel matrix. Different choices of kernel functions are appropriate for different ML applications. Two of the more commonly used kernel functions are,

$$\text{Gaussian}: \; k(x_i,x_j)=e^{-\|x_i-x_j\|_2^2/(2\sigma^2)} \tag{9}$$

$$\text{Laplacian}: \; k(x_i,x_j)=e^{-\|x_i-x_j\|_1/\sigma} \tag{10}$$

defined as using the Euclidean ($L^{[2]}$) and Manhattan ($L^{[1]}$) norm, respectively. Here, σ represents the kernel width, a measure of locality in the used training set. Without any loss of generality,

we restrict ourselves to the Laplacian kernel for this study. This choice is motivated by the fact that it has shown good performance in the context of predicting molecular atomization energies.[2] However, the Gaussian, or other kernels, could have been used just as well.

As a relevant benchmark of performance of the ML model with a given representation, we use the error in predicting new data outside **x**, the generalization error (GE). There are several ways to estimate the GE. We exclude some subset of all available data **x** (and **y**) from the solution of Eq. (8) and then evaluate the mean absolute error (MAE) for the prediction of the excluded data. The subset of **x** used in Eq. (8) is called the *training set* $\mathbf{x}_{\text{train}}$, and the remaining input the *test set* $\mathbf{x}_{\text{test}}$. Hence, we define

$$\text{MAE}_{\text{test}} = \frac{1}{m} \sum_{i \in \mathbf{x}_{\text{test}}} |y_i - f(x_i, \boldsymbol{\alpha})|, \tag{11}$$

where m is the size of the test set. The measure MAE_{test} determines how well the machine predicts new data. The number of systems included in $\mathbf{x}_{\text{test}}$ needs to be sufficiently large for this measure to be an accurate estimate of the GE. Hence, there is a trade-off between using available data as part of $\mathbf{x}_{\text{test}}$ or $\mathbf{x}_{\text{train}}$. We alleviate this problem by a statistical validation method known as k-fold cross-validation following the recipes in Ref. [2], combined with random sampling. The MAE_{test} is calculated as the mean of the MAE_{test} of several independent ML runs. In each run, $\mathbf{x}_{\text{test}}$ is constructed by selecting a specific number of entries randomly out of the full dataset.

Similarly, $\text{MAE}_{\text{train}}$ is defined by substituting $\mathbf{x}_{\text{train}}$ for $\mathbf{x}_{\text{test}}$ in Eq. (11). This measure determines the precision with which the machine reproduces the data it has been trained on. A high $\text{MAE}_{\text{train}}$ suggests the machine has too few degrees of freedom to describe the data well. A low $\text{MAE}_{\text{train}}$ suggests high flexibility of the fitting function, a potential risk of overfitting, and that it may be possible to reach good transferability with a sufficiently large dataset.

Representations

The feature vector representation used for the input is of central importance for the ML scheme. Here, this is the mapping of the positions and identities of all atoms into a vector x. We expect a well-suited representation to exhibit the following beneficial properties (with examples in parenthesis given for molecules).

1. *Complete, nondegenerate:* x should incorporate all features in the input that are relevant for the underlying problem. (Two different molecules should give different x.)
2. *Compact, unique:* x should have minimal features that are redundant to the underlying problem. (Two instances of the same molecule, but rotated differently, should give the same x.)
3. *Descriptive:* instances of input that are "close," giving similar y, should generally be represented by x that are close, in the sense of a small $||x_1 - x_2||$. (Two molecules that are identical except for small differences in the atomic positions, which have similar y, should generally have similar x.)
4. *Simple:* generating the representation for a given input should require as little computational effort as possible. (The set of eigenvalues from solution of the many-body Schrödinger equation of the molecule would generally not be a useful x.)

A bijective representation is perfectly nondegenerate and unique, that is, it has a one-to-one correspondence between the input and x. In this discussion, the distinction made by "relevant for the underlying problem" is important, because different features of the input may be relevant for different applications. For example, in ML models of atomization energies the chirality of a molecule is not relevant, but in ML models of the optical activity in circular dichroism it is.

A common method for analysis of the properties of different feature vector representations is principal component analysis (PCA).[20] This method reduces the dimensionality of a dataset while still conserving as much information as possible. To generate the PCA, a singular value decomposition[21] is performed on **x**,

$$\mathbf{x} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{W}^{\mathrm{T}}. \tag{12}$$

Submatrices are created from the first k columns extracted from the resulting matrices, $\mathbf{U}_k$ and $\boldsymbol{\Sigma}_k$. The PCA dataset of reduced rank k is then defined as

$$\mathbf{z} = \mathbf{U}_k\boldsymbol{\Sigma}_k. \tag{13}$$

This dataset is not only useful for visualization. If one suspects that a feature vector representation is not sufficiently compact (in the sense of point 2 in the list of beneficial properties given above in this section), one can try to replace **x** with the PCA feature vector representation, **z**, to extract a representative lower-dimensional part. Note that the PCA representation should only be constructed on the training set. The test set may not be used.

Molecular coulomb matrix

This work aims at extending the ML model for molecular properties[1,2,5] to properties of crystals. We, therefore, briefly summarize the molecular approach in the following. The feature vector representation called the *Coulomb matrix* is a symmetric atom by atom matrix given in Hartree atomic units,

$$x_{ij} = \begin{cases} 0.5 Z_i^{2.4} & \text{if } i = j \\ Z_i Z_j \phi(||\mathbf{r}_i - \mathbf{r}_j||_2) & \text{if } i \neq j \end{cases} \tag{14}$$

$$\phi(r) = 1/r \tag{15}$$

where $\phi(r)$ is the Coulomb potential, Z_i and $\mathbf{r}_i$ are the atomic number, and position of the ith atom. The nondiagonal elements correspond thus to the pair-wise Coulomb repulsion

between the positive atomic cores in the system, and the diagonal elements are chosen by construction as the result of an exponential fit to the potential energy of a free atom.

While this representation is not bijective, it is nondegenerate in the sense that no two molecules that differ more from each other than being enatiomers will yield the same Coulomb matrix. One molecule, however, can result in several Coulomb matrices due to the various ways of ordering atoms. Atom index invariance can be introduced by ordering atom indices according to the norm of each row (or column),[1,2] or using permuted sets of Coulomb matrices.[5] Sorting, however, introduces the issue of differentiability—a potentially desirable property when it comes to the modeling of atomic forces. Use of the (differentiable) eigenvalue spectrum of the Coulomb matrix yields an atom index invariant descriptor at the expense of losing uniqueness.[22,23] There is another atom index invariant and differentiable representation, the Fourier series of atomic radial distribution functions, that also preserves uniqueness as well as spatial invariances.[8] However, this representation has not yet been explored in depth and has, therefore, not been included in the representations discussed herewithin.

To benchmark our results for solids, we compare to the performance for the molecular ML model that has been trained and tested on (subsets of) the GDB-database in previous studies.[1,24–26] We use a subset of 7165 entries out of the GDB-13 dataset. These entries are all the organic molecules in this set with up to seven atoms of elements C, O, N, and S, and valencies satisfied by hydrogen atoms. The atomic coordinates were relaxed using atomic force fields, and the atomization energies calculated with a higher-order first-principles method, density functional theory[27,28] using the PBE0 hybrid functional.[29–31] We call this dataset QM7.[24–26,32]

When considering ways of extending the Coulomb matrix to periodic systems, one might be tempted to take the Coulomb matrix for just the atoms in the primitive unit cell of the periodic crystal, alongside with the unit cell vectors. However, depending on the choice of primitive unit cell the set of interatomic distances will vary as distances to neighboring atoms from different unit cells are not accounted for. Hence, such a representation is not unique in the sense that the same crystal can lead to different representations, and therefore, is less likely to yield well performing ML models.[4]

Ewald sum matrix

We now consider a straightforward extension of the Coulomb matrix representation that removes the most obvious dependence on the nonunique set of interatomic distances in the primitive unit cell. We form an atom by atom matrix with one element for each pair of atoms in the primitive unit cell, but now each element is defined to represent the full Coulomb interaction energy corresponding to all infinite repetitions of these two atoms in the lattice. In this way, the elements in the matrix retain essentially the same meaning as in the Coulomb matrix while the complete infinite repetition of the lattice is taken into account. As such, one can propose the following expression for the matrix elements,

$$x_{ij} = \frac{1}{N} Z_i Z_j \sum_{k,l,k\neq l} \varphi(||\mathbf{r}_k - \mathbf{r}_l||_2) \tag{16}$$

where the sum over k is taken over the atom i in the unit cell and its N closest equivalent atoms, and similarly for l and j. The intention is to take $N \to \infty$ to represent the full electrostatic interaction between the infinitely repeated atoms equivalent to atoms i and j in the primitive cell. However, this type of infinite electrostatic sum has well-known issues with convergence that have been discussed at length in the field of materials science. One resolution is given by the Ewald sum.[33–35] The central idea is to divide the problematic double sum in Eq. (16) into two rapidly converging sums and one constant,

$$x_{ij} = x_{ij}^{(r)} + x_{ij}^{(m)} + x_{ij}^{0}, \tag{17}$$

where $x_{ij}^{(r)}$ is the short range interaction calculated in real space, $x_{ij}^{(m)}$ the long-range interaction calculated in the reciprocal space, and x_{ij}^{0} is a constant. The division is controlled by a screening length parameter a, which influences how rapidly the sums converge. The first term is given by

$$x_{ij}^{(r)} = Z_i Z_j \sum_{\mathbf{L}} \frac{\mathrm{erfc}(a||\mathbf{r}_i - \mathbf{r}_j + \mathbf{L}||_2)}{||\mathbf{r}_i - \mathbf{r}_j + \mathbf{L}||_2} \quad (i \neq j), \tag{18}$$

where the sum is taken over all lattice vectors $\mathbf{L}$ inside a sphere of a radius set by a cutoff $L_{\max}$. The second term is

$$x_{ij}^{(m)} = \frac{Z_i Z_j}{\pi V} \sum_{\mathbf{G}} \frac{e^{-||\mathbf{G}||_2^2/(2a)^2}}{||\mathbf{G}||_2^2} \cos(\mathbf{G} \cdot (\mathbf{r}_i - \mathbf{r}_j)) \quad (i \neq j), \tag{19}$$

taken over all non-zero reciprocal lattice vectors $\mathbf{G}$ in a sphere of radius set by a cutoff $G_{\max}$, taken to be large enough for the sum to converge; and V is the unit cell volume. The last term is

$$x_{ij}^{0} = -(Z_i^2 + Z_j^2)\frac{a}{\sqrt{\pi}} - (Z_i + Z_j)^2 \frac{\pi}{2Va^2} \quad (i \neq j), \tag{20}$$

where the first term in Eq. (20) is the Ewald self-terms for the i and j sites and the second term is a correction needed as we use a charged cell, as in analog to the Coulomb matrix representation, the expressions describe the interaction from the positive atomic cores. The correction makes the total energy be that of a system with a uniform compensating background that makes the system neutral. For the diagonal terms in the matrix ($i = j$), we take the Ewald sum interation energy of the lattice of i-type atoms, which is given by the same Eqs. (18) and (19) but with an extra factor 1/2 and,

$$x_{ii}^{0} = -Z_i^2 \frac{a}{\sqrt{\pi}} - Z_i^2 \frac{\pi}{2Va^2}. \tag{21}$$

The value of a only affects the rate of convergence in the above sums, not the final value of x_{ij}. There are several suggested schemes for how to set it, in our work we take

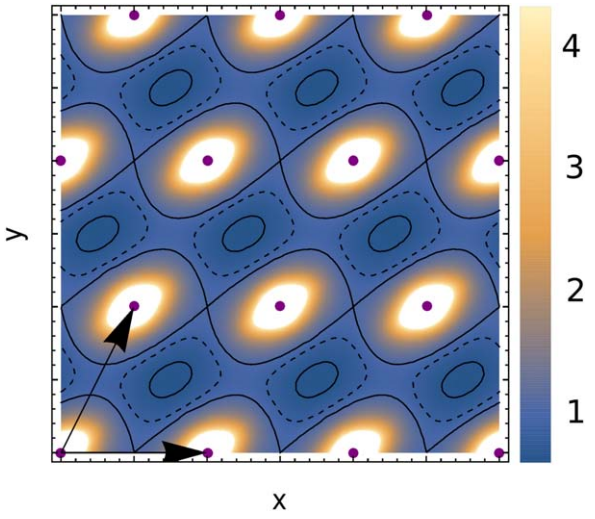

Figure 1. An illustration of $\tilde{\Phi}(\mathbf{r}_1, \mathbf{r}_2)$ used in the sine matrix representation, Eq. (24), for a 2D crystal lattice in a primitive unit cell shown by arrows. The figure shows the magnitude of our constructed "interaction" between one atom at $\mathbf{r}_1=(x,y)$ and another fixed at the origin $\mathbf{r}_2=0$ (the latter shown along with its infinite repetitions as solid purple dots.) The interaction is periodic across the unit cells and grows to infinity as $\mathbf{r}_1$ approach any repetition of the atom at the origin.

$$a=\sqrt{\pi}\left(\frac{0.01M}{V}\right)^{1/6} \tag{22}$$

where M is the number of atoms in the unit cell.

We refer to Eqs. (18)–(20) as the *Ewald sum matrix* representation. Our definitions are chosen to make each element of the matrix be the full Ewald sum of the Coulomb interaction between the sites i and j (or i with itself). We note briefly that the python materials genomics (pytmatgen) open source library[36] contains a similar matrix as an intermediate step in the calculation of the full Ewald sum energy. However, that matrix differ from ours in that it is defined to make the sum over all elements give the total energy in a way that makes individual matrix elements depend on the specific value of the a parameter used. This seems an undesirable feature for using the matrix as a descriptor. (Nevertheless, for the value a in our calculations, we use the same formula as in pymatgen, Eq. (22).)

Extended coulomb-like matrix

Another way to generalize the Coulomb matrix to periodic systems is to extend the size of the representation matrix. Let each element be the electrostatic interaction between one of the M atoms in the unit cell and one of the atoms in the N closest unit cells, giving an M by $N \cdot M$ matrix representation on the regular Coulomb matrix form of Eq. (14), with $1 \leq i \leq M$, $1 \leq j \leq N \cdot M$. To completely avoid the dependence on the chosen primitive unit cell, one would like to take $N \to \infty$, that is, an infinitely large representation matrix. This can be avoided if the long-range electrostatic interaction is replaced with a more rapidly decaying interaction. Here, we chose $\tilde{\varphi}(r)=e^{-r}$. In this way, the elements of the matrix quickly drop to zero, and the representation matrix can be cut off at a finite dimension that is taken to be sufficiently large for all systems in the dataset. We refer to this as the *extended Coulomb matrix* representation. One benefit of this representation over the Ewald sum matrix is that it is more straightforward to evaluate (one can simply iterate over N copies of the unit cell).

Sine matrix

Yet another representation can be constructed by further extending the idea of reducing the computational effort by replacing the long-range electrostatic interaction by a simpler expression. We start from the M by M Ewald sum matrix in Eq. (16), but substitute the whole sums of the electrostatic interaction with an arbitrarily chosen two-point potential that is intended to share the same basic properties as such a sum, giving

$$x_{ij}=\begin{cases} 0.5Z_i^{2.4} & \text{if } i=j \\ Z_iZ_j\tilde{\Phi}(\mathbf{r}_i,\mathbf{r}_j) & \text{if } i\neq j \end{cases} \tag{23}$$

where $\mathbf{r}_I$ is the position of the I^{th} atom in the unit cell.

Consider two nonequivalent atoms in the unit cell, A and B. The Coulomb sum contribution due to the infinitely repeated grid of A and B atoms can be thought of as a potential field which is a function of the position of the atom A. Three important properties of this field are: (i) the expression as function of each atomic coordinate is periodic with respect to the crystal lattice; (ii) the contribution from two equivalent atoms in neighboring cells should be the same; (iii) the potential should approach infinity when A takes the same position as B. The conclusion is that the potential needs to be symmetric with respect to the lattice vectors.

A possible choice for $\tilde{\Phi}(\mathbf{r}_1, \mathbf{r}_2)$ that fulfills these requirements is

$$\tilde{\Phi}(\mathbf{r}_1,\mathbf{r}_2)=||\mathbf{B}\cdot\sum_{k=\{x,y,z\}}\hat{e}_k\sin^2[\pi\hat{e}_k\mathbf{B}^{-1}\cdot(\mathbf{r}_1-\mathbf{r}_2)]||_2^{-1} \tag{24}$$

where $\hat{e}_x, \hat{e}_y, \hat{e}_z$ are the coordinate unit vectors and $\mathbf{B}$ is the matrix formed by the basis vectors of the lattice. The product inside the sine function thus gives the vector between the two sites expressed in crystal lattice coordinates, which gives the right periodicity in $\mathbf{r}_1$ and $\mathbf{r}_2$. We call Eqs. (23) and (24) the *sine matrix* representation. The benefit of this representation over the others suggested in this work is that Eq. (23) is a completely straightforward M by M matrix that only depends on the positions of the atoms in a single unit cell. Hence, the computational load of this representation is minimal. Figure 1 shows $\tilde{\Phi}$ for a two-dimensional (2D) lattice.

Note that we do not have a proof for the completeness or uniqueness of these representations. They are merely constructed as sensible extensions of the Coulomb matrix idea.

Implementation Details and Datasets

We have implemented the KRR ML scheme using a Laplacian kernel both for the original set of molecules from Ref. [1] with

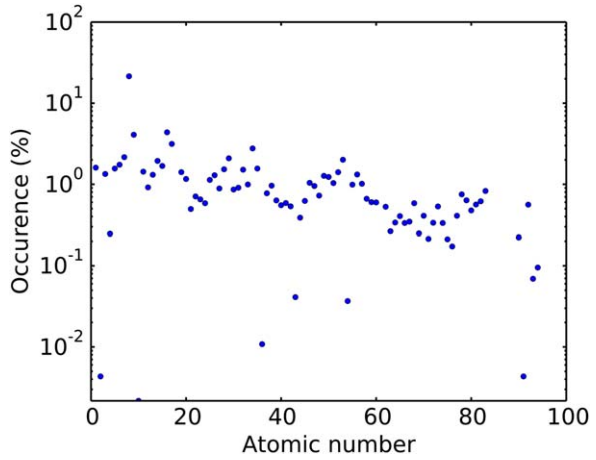

Figure 2. The frequency of occurrence of various elements in the 3938 systems in the MP dataset. The distribution is not uniform, but rather expected to reflect the occurrence of elements in published materials. The four most common elements are O, Si, Cu, and S.

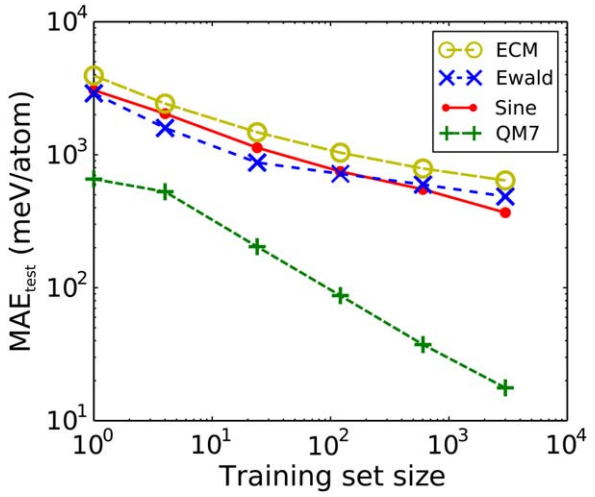

Figure 3. The mean absolute GE, Eq. (11), versus training set size for the different representations considered in this work. Shown are MAE_{test} for predicting formation energies of crystals in the MP dataset: (Sine) Eqs. (23) and (24); (Ewald) Eqs. (18–20); and (GCM) described in extended coulomb-like matrix section. For comparison, we also include (QM7), MAE_{test} for atomization energies of molecules in the QM7 dataset using a regular Coulomb matrix, Eqs. (14) and (15).

the Coulomb matrix representation, and for the discussed representations using a dataset of crystal structures. We use the Python programming language, including numpy,[37] scipy,[38] and pytmatgen.[36]

Our dataset of formation energies of periodic solids was obtained from the materials project (MP) database.[39] We extracted 3938 systems from the MP without obvious order. As the MP database is derived from the ICSD,[40,41] the distribution of elements in the extracted systems roughly match their occurrence in published materials in the literature. This distribution is shown in Figure 2. While this may be interpreted as a bias in the dataset, one can also see it as representative for the intended application of ML in predicting properties of materials out of available material databases based on published materials.

The machine requires values for the regularization parameter λ and kernel width σ. We follow Hansen et al.;[2] optimal values were identified by calculating MAE_{test} for a training set size of 3000 for pairs of λ and σ values on a 2D logarithmic grid, using a spacing factor of 2 for σ and 10 for λ. (This method works well when the model has a small number of hyper parameters, but needs to be replaced with more sophisticated methods for ML models with a larger set of parameters since the time it takes to find the optimum is of $\mathcal{O}(x^p)$ where p is the number of hyper parameters.) The optimal values found are shown in Table 1. The MAE_{test} for the different representations is not very sensitive to these parameters, that is, the regions around the minimums in the generated grids were relatively flat.

Table 1. Optimal values of parameters λ and σ for the machines using the different feature vector representations in this work, and for a training set consisting of 3000 crystals drawn at random from the MP.

Representation	σ	λ
Ewald sum matrix	$1.0 \cdot 10^5$	10^{-4}
Generalized Coulomb matrix	$8.0 \cdot 10^4$	10^{-3}
Sine matrix	$4.0 \cdot 10^4$	10^{-4}
Coulomb matrix for QM7 molecules	$2.5 \cdot 10^3$	10^{-6}

Results

The performance of the representations of periodic systems studied in this work are meant to be considered in the context of the excellent performance of the machine for molecules presented in Refs. [1,2]. To make this comparison clear, we have reproduced the GE of their scheme with our implementation and the QM7 dataset. The results are shown in Figure 3. In Figure 4, a 2D PCA of the QM7 set using the Coulomb matrix representation is shown. Already with a training set size of a couple of hundred atoms, the MAE approaches the accuracy of DFT with the least computationally expensive, semilocal, functionals. At a training set size of 500, we arrive at a MAE_{test} of around 0.07 eV/atom, whereas DFT with the functional by Perdew, Burke, and Ernzerhof (PBE)[29] has an MAE for the atomization energy of small molecules of about 0.15 eV/atom.[29] As demonstrated by Rupp et al.,[1] the precision of the predictions of the machine keeps improving with increased size of the training set, and at 3000 structures one finds an GE error below 0.02 eV/atom.

We now turn to the results of applying ML to periodic crystals using the feature vector representations in this work, shown in Figure 3. The performance of all our representations is similar, but their accuracy is inferior to what we see for molecules. All the representations studied improve greatly with increased training set size. The MAE_{test} for the representations at a training set size of 3000 are 0.49 eV/atom for the Ewald sum matrix, 0.64 eV/atom for the extended Coulomb matrix, and 0.37 eV/atom for the sine matrix representation, that is, an order of magnitude worse than we find for the molecules. Furthermore, we find that MAE_{train} for all the representations are insignificant ($<5 \cdot 10^{-3}$) even at a training set size of 3000.

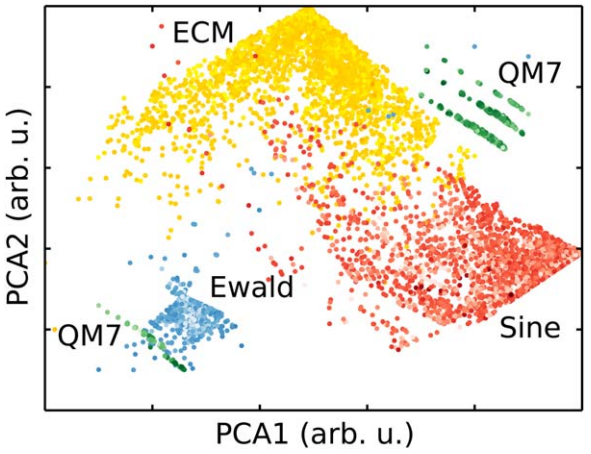

Figure 4. A 2D PCA of the datasets used in this work: (QM7, green) the Coulomb matrix representation of the QM7 dataset of molecules; (Sine, red) the sine matrix representation using the MP dataset; (Ewald, blue) the Ewald sum matrix representation using the MP dataset; (ECM, yellow) The extended Coulomb matrix dataset using the MP dataset. The PCA of the representations of the MP dataset differ substantially from the PCA of the Coulomb matrix of QM7. The shading of the points represent the target value energy, showing that there is no clear energy-PCA pattern.

Figure 4 compares the 2D PCA of the MP dataset to the QM7 one for the different representations. The QM7 PCA is localized to a set of small clusters. The MP data for the sine and the extended Coulomb matrix representations appear more uniformly spread out. The Ewald sum matrix PCA has a central cluster but also a few data points far removed from this cluster.

Discussion

The results in Figure 3 suggest that the sine matrix representation is slightly better than the other representations in this work, in that it reaches a lower GE and has better development of the GE with training set size. However, the performance of the different methods are roughly similar, indicating that the selection between the feature vector representations presented in this work for a given application may not have major impact on the GE. This further strengthens the case of the sine matrix representation, as it is the representation that requires the least computational expense. It is interesting to note that while the PCAs of the sine and extended Coulomb matrix bear strong resemblance, the Ewald sum PCA is distinctly different with a strong clustering and a few outlier points. Furthermore, the GE of all representations systematically decreases with increasing training set size. These results suggest the GE could be reduced even further if only more extensive dataset for training were available.

When comparing the molecular results with the ones for solids, it is important to keep in mind that the diversity and composition of the MP dataset differs substantially from that of QM7. In particular, only very few element types are present in the molecular dataset consisting of atoms with very finite size, no molecule has more than seven atoms (not counting hydrogens). This is illustrated by the PCA in Figure 4, where the QM7 data is collected in a few tight clusters. Arguably, the chemical space of QM7 is significantly smaller than the one we use to evaluate the representations for solids as the MP dataset contains 10 times more elements than QM7.

Hence, the central conclusion of the present work is that there are three areas where the situation of ML for periodic systems can be improved: (i) our methods should be used with even larger datasets to confirm that the GE can be brought down to levels where it is useful for applications, (ii) further improved representations may be helpful if they more efficiently can represent the degrees of freedom offered by periodic crystals. Such representations may reduce the need for larger training sets; and (iii) if a way of generating a dataset over a restricted chemical space can be devised which is as compact as QM7, we may reach more promising ML performance for periodic systems. We are presently working in all of these directions.

Summary and Conclusions

We have investigated the performance of several crystal structure representations for ML models of formation energies of solids. Our work is a natural generalization of an ML scheme previously shown to be successful for the atomization energy of molecules. We have compared three different representations, and found that a sine matrix which simulates the features of an infinite Coulomb sum is both most efficient, and gives the smallest GE error. While the performance of all the methods may at first seem disappointing when compared with the small GE confirmed for molecules, we can explain this discrepancy to be due to datasets which are too small to cover the hugely diverse compositional and structural space with sufficient density. As such, the full potential of ML for periodic systems still has to be demonstrated. The improvement of the MAE_{test} with training set size suggests, however, that the methods presented here can lead to accurate machine models if only trained on larger or more restricted data subsets. Hence, our results offer promising indications that sufficiently accurate and transferable ML models of energies of periodic systems can be realized.

Acknowledgments

Calculations have been performed at the Swedish National Infrastructure for Computing (SNIC). This research used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory.

Keywords: machine learning · formation energies · representations · crystal structure · periodic systems

How to cite this article: F. Faber, A. Lindmaa, O. A. von Lilienfeld, R. Armiento *Int. J. Quantum Chem.* **2015**, *115*, 1094–1101. DOI: 10.1002/qua.24917

[1] M. Rupp, A. Tkatchenko, K.-R. Müller, O. A. von Lilienfeld, *Phys. Rev. Lett.* **2012**, *108*, 058301.

[2] K. Hansen, G. Montavon, F. Biegler, S. Fazli, M. Rupp, M. Scheffler, O. A. von Lilienfeld, A. Tkatchenko, K.-R. Müller, *J. Chem. Theory Comput.* **2013**, *9*, 3404.
[3] O. A. von Lilienfeld, *Int. J. Quantum Chem.* **2013**, *113*, 1676.
[4] K. T. Schütt, H. Glawe, F. Brockherde, A. Sanna, K. R. Müller, E. K. U. Gross, *Phys. Rev. B* **2014**, *89*, 205118.
[5] G. Montavon, M. Rupp, V. Gobre, A. Vazquez-Mayagoitia, K. Hansen, A. Tkatchenko, K.-R. Müller, O. A. von Lilienfeld, *New J. Phys.* **2013**, *15*, 095003.
[6] A. Lopez-Bezanilla, O. A. von Lilienfeld, *Phys. Rev. B* **2014**, *89*, 235411.
[7] L.-F. Arsenault, A. Lopez-Bezanilla, O. A. von Lilienfeld, A. J. Millis, *Phys. Rev. B* **2014**, *90*, 155136.
[8] O. A. von Lilienfeld, R. Ramakrishnan, M. Rupp, A. Knoll, *Int. J. Quantum Chem.* **2015**, *115*, 1084.
[9] A. P. Bartók, R. Kondor, G. Csányi, *Phys. Rev. B* **2013**, *87*, 184115.
[10] B. Meredig, A. Agrawal, S. Kirklin, J. E. Saal, J. W. Doak, A. Thompson, K. Zhang, A. Choudhary, C. Wolverton, *Phys. Rev. B* **2014**, *89*, 094104.
[11] G. Pilania, C. Wang, X. Jiang, S. Rajasekaran, R. Ramprasad, *Sci. Rep.* **2013**, *3*, 2810.
[12] A. P. Bartók, M. J. Gillan, F. R. Manby, G. Csányi, *Phys. Rev. B* **2013**, *88*, 054104.
[13] A. E. Hoerl, R. W. Kennard, *Technometrics* **1970**, *12*, 55.
[14] V. Vovk, In Empirical Inference; B. Schölkopf, Z. Luo, V. Vovk, Eds.; Springer: Berlin, Heidelberg, **2013**; pp. 105–116.
[15] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed.; Springer: New York, **2011**.
[16] K. Muller, S. Mika, G. Ratsch, K. Tsuda, B. Scholkopf, *IEEE Trans. Neural Netw.* **2001**, *12*, 181.
[17] V. N. Vapnik, The Nature of Statistical Learning Theory; Springer-Verlag: New York, **1995**.
[18] B. Schölkopf, A. J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond; The MIT Press: Cambridge, Mass, **2001**.
[19] B. Schölkopf, A. Smola, K.-R. Müller, *Neural Comput.* **1998**, *10*, 1299.
[20] I. T. Jolliffe, Principal Component Analysis; Springer: New York, **2002**.
[21] P. G. H. Golub, D. C. Reinsch, *Numer. Math.* **1970**, *14*, 403.
[22] J. E. Moussa, *Phys. Rev. Lett.* **2012**, *109*, 059801.
[23] M. Rupp, A. Tkatchenko, K.-R. Müller, O. A. von Lilienfeld, *Phys. Rev. Lett.* **2012**, *109*, 059802.
[24] L. C. Blum, J.-L. Reymond, *J. Am. Chem. Soc.* **2009**, *131*, 8732.
[25] T. Fink, H. Bruggesser, J.-L. Reymond, *Angew. Chem. Int. Ed.* **2005**, *44*, 1504.
[26] T. Fink, J.-L. Reymond, *J. Chem. Inf. Model.* **2007**, *47*, 342.
[27] P. Hohenberg, W. Kohn, *Phys. Rev.* **1964**, *136*, B864.
[28] W. Kohn, L. J. Sham, *Phys. Rev.* **1965**, *140*, A1133.
[29] J. P. Perdew, K. Burke, M. Ernzerhof, *Phys. Rev. Lett.* **1996**, *77*, 3865.
[30] M. Ernzerhof, J. P. Perdew, K. Burke, *Int. J. Quantum Chem.* **1997**, *64*, 285.
[31] M. Ernzerhof, G. E. Scuseria, *J. Chem. Phys.* **1999**, *110*, 5029.
[32] R. Ramakrishnan, P. O. Dral, M. Rupp, O. A. von Lilienfeld, *Sci. Data* **2014**, *1*, 140022.
[33] P. P. Ewald, *Ann. Phys. (Berlin)* **1921**, *369*, 253.
[34] A. Y. Toukmaji, J. A. Board, Jr., *Comput. Phys. Commun.* **1996**, *95*, 73.
[35] R. M. Martin, Electronic Structure: Basic Theory and Practical Methods; Cambridge University Press: Cambridge, **2008**.
[36] S. P. Ong, W. D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V. L. Chevrier, K. A. Persson, G. Ceder, *Comput. Mater. Sci.* **2013**, *68*, 314.
[37] S. van der Walt, S. Colbert, G. Varoquaux, *Comput. Sci. Eng.* **2011**, *13*, 22.
[38] E. Jones, T. Oliphant, P. Peterson, and others, SciPy: Open source scientific tools for Python; http://www.scipy.org/ (accessed on Jan 23, 2015).
[39] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, K. A. Persson, *APL Mater.* **2013**, *1*, 011002. https://materialsproject.org/ (accessed on Jan 23, 2015).
[40] G. Bergerhoff, R. Hundt, R. Sievers, I. D. Brown, *J. Chem. Inf. Comput. Sci.* **1983**, *23*, 66.
[41] A. Belsky, M. Hellenbrandt, V. L. Karen, P. Luksch, *Acta Crystallogr. Sect. B Struct. Sci.* **2002**, *58*, 364.

Received: 30 January 2015
Revised: 26 March 2015
Accepted: 30 March 2015
Published online 20 April 2015