doi:10.2533/chimia.2019.972 Chimia 73 (2019) 972–982

Machine Learning at the Atomic Scale

Félix Musil and Michele Ceriotti*

Abstract: Statistical learning algorithms are finding more and more applications in science and technology. Atomic-scale modeling is no exception, with machine learning becoming commonplace as a tool to predict energy, forces and properties of molecules and condensed-phase systems. This short review summarizes recent progress in the field, focusing in particular on the problem of representing an atomic configuration in a mathematically robust and computationally efficient way. We also discuss some of the regression algorithms that have been used to construct surrogate models of atomic-scale properties. We then show examples of how the optimization of the machine-learning models can both incorporate and reveal insights onto the physical phenomena that underlie structure–property relations.

Keywords: Machine learning

Félix Musil was born in Paris (France) in 1991. He studied physics at the EPFL and received his MSc in applied physics in 2015, with a thesis on the modelling of plasma in a fusion reactor. For his PhD he joined the group of Prof. Ceriotti at the EPFL to develop and apply methods to investigate structure–property relationships in materials using atomistic modelling and machine learning techniques.

Michele Ceriotti received his PhD in Physics from ETH Zürich. He spent three years in Oxford as a Junior Research Fellow at Merton College. Since 2013 he leads the laboratory for Computational Science and Modeling in the Institute of Materials at EPFL. His research revolves around the atomic-scale modelling of materials, based on the sampling of quantum and thermal fluctuations and on the use of machine learning to predict and rationalize structure-property relations. He has been awarded the IBM Research Forschungspreis in 2010, the Volker Heine Young Investigator Award in 2013, an ERC Starting Grant in 2016, and the IUPAP C10 Young Scientist Prize in 2018.

1. Introduction

The steady increase in computing power in the last decades, together with the improvements in accuracy and efficiency of electronic structure methods and empirical force fields (FFs), have given atomistic modeling a central role in the investigation of molecular and condensed-phase systems, and underpinned the rise of computational material design. Some recent achievements include the study of synaptic transmission mechanisms,[1] water splitting with photo-electrical cells,[2] realistic metal deformations and plasticity[3] and nucleation with billions of atoms.[4] Nevertheless the inherent scaling of *ab initio* methods limits their applicability, preventing systems with more than a few thousand atoms from being studied, while the development of accurate and transferable reactive, multi-component empirical FFs remains a major challenge. The last decade has seen the emergence of machine learning (ML) methods in the field of atomic-scale modeling to automate time consuming analyses[5–7] (unsupervised learning) or to reduce the cost of predicting quantities associated with atomic systems[8–11] (supervised learning). Unsupervised techniques aim at unravelling patterns in databases, which in the context of atomistic modeling can correspond to identifying recurring motifs within structures,[12,13] as well as groups of 'similar' structures in datasets of molecules and molecular solids[14,15] or molecular dynamics trajectories.[16–18] Given a set of atomic structures $\{\mathcal{A}_n\}$ associated with some properties $\{\boldsymbol{y}_n\}$, *e.g.* energy or other observables computed by electronic structure theory, supervised ML methods can be used to learn a surrogate model $F : \mathcal{A} \to \boldsymbol{y}$ to predict those properties. In this way, ML makes it possible to bypass solving Schrödinger's equation, and to obtain inexpensive and accurate predictions of the formation energy of atomic structures,[19,20] the chemical shieldings in molecular materials,[21] the electron density of small molecules,[22,23] the electron transfer coupling between dimers[24] *etc.* One of the most promising applications for these algorithms is to provide frameworks to systematically build accurate interatomic potentials[25–27] for a slightly higher running cost than traditional FFs.

In this review, we briefly summarize some of the approaches that have been used to model atomic scale properties with ML techniques. We begin by providing a detailed discussion of the problem of obtaining a representation of atomic configurations, *i.e.* how the Cartesian coordinates of the atoms can be transformed to obtain a mathematical description of the structure that is concise, and that incorporates the fundamental physical symmetry. In doing so, we will show how most of the existing representations can be seen as different views of a symmetrized atomic density. We then give a brief overview of the regression techniques that have been used in the context of atomic-scale modeling, focusing in particular on Gaussian process regression, and discussing some of the aspects that are particularly relevant in the learning of atomic-scale properties. Finally we show how representations and regression models can be improved by incorporating more prior knowledge about the specific problem, using recent applications to highlight some of their key features.

*Correspondence: Prof. Dr. M. Ceriotti, E-mail: michele.ceriotti@epfl.ch
Laboratory of Computational Science and Modeling, IMX, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne

2. Atomic-scale Representations

The rise of ML during the last ten years has been mostly fueled by the emergence of models able to learn relevant features from raw data, *e.g.* images, texts, *etc.*, alongside the parameters needed to perform tasks such as detecting objects or translating sentences.[28] In this context, deep-learning models that simply treat data as a stream – or an array – of bytes have outperformed models incorporating knowledge about the grammar of a language, or the content of a set of pictures.[29] Unlike many computer science applications, the properties of a physical system obey a number of symmetries and conservation laws, and efforts to encode these at the core of atom-scale models of matter have been shown to consistently improve the data efficiency of the regression scheme, making better use of the expensive electronic-structure calculations used for training. One option is to incorporate symmetries at the level of the model. For example, extensions of the CNN architecture to extract invariant and/or covariant features from 3D shapes like an atomic structure[30–35] have been recently developed. The main approach followed in the atomic scale modeling community this far has however been to develop representations of the atomic structure that are equivariant with respect to these symmetries. Using these features as the input representation gives a ML model adapted to the desired symmetries.

Several authors have proposed to represent structures in terms of so-called fingerprints by concatenating features associated with an atomic structure, *e.g.* elemental properties, atomic connectivity, electronic structure attributes, stoichiometry, *etc*.[9,36–38] to build models for complex properties such as melting temperature, dielectric constant and band gap energy. While in principle any feature can be introduced into a ML model, electronic structure theory shows that any ground-state property of a structure $\mathcal{A}$ is a smooth function of the set of N atomic coordinates $\{\mathbf{r}_i\}$ and chemical species $\{\alpha_i\}$.[39] These considerations suggest that representations of the atomic structure based only on this core information provide a physically-motivated basis to regress any property $y(\mathcal{A})$ that could be computed by solving the Schrödinger equation for the structure.

While a representation in terms of $\{\mathbf{r}_i, \alpha_i\}$ provides a complete description of a structure $\mathcal{A}$, it does not incorporate the most basic physical symmetries that could follow a property, such as the invariance to the labelling of identical nuclei, or rigid translations and rotations of the reference frame. Many schemes have been proposed in recent years to translate the essential inputs of a quantum calculation code into a representation that incorporates these symmetries, and that can then be used in combination with most regression algorithms to learn physical properties in a data-efficient manner. Some start from internal coordinates of a molecule, such as the distances and angles between atoms,[19,27,40–45] that are rotationally and translationally invariant while others begin with an atomic density[44,46–51] which is invariant under the permutation of the atom indices. As we illustrate below, many of these representations have been shown to be essentially equivalent, as they correspond to special cases of a general framework generating invariant and covariant representations from atomic densities.[47,52,53] In the following text, we focus on local invariant representations but this framework is also a powerful tool to develop local covariant representations,[52,54,55] as well as representations that capture non-local, global features of a given structure.[14,56]

We emphasize the generality and abstract nature of this construction by associating with each structure a vector $|\mathcal{A}\rangle$. Different representations can be thought of as resulting from particular choices of the basis that is used to provide a concrete protocol to evaluate $|\mathcal{A}\rangle$ much like the wavefunction can be expressed equally well in real space, in plane waves, or in one of the many localized basis sets that have been used in quantum chemistry. We choose a real-space basis as the starting point, and associate with $|\mathcal{A}\rangle$ a set of element-resolved smooth atomic densities

$$\langle \alpha \mathbf{r} | \mathcal{A} \rangle = \sum_{i \in \mathcal{A}, \alpha} g\left(\mathbf{r} - \mathbf{r}_i\right). \tag{1}$$

The sum extends over all atoms of type α within the structure, and g is a smooth density function (a function peaked at zero with central symmetry that decreases to zero smoothly). The use of a smooth density function instead of a Dirac distribution to represent the atomic coordinates ensures that the resulting representation is smooth with respect to atomic displacements. Provided that the functions g are sufficiently peaked, this representation determines fully the position of all the atoms, and is clearly independent on the order in which atoms are considered.

It is, however, not invariant with respect to rotations and translations. These additional symmetries can be incorporated through Haar integration[57] of the atomic density, *i.e.* averaging over the corresponding group

$$|\mathcal{A}\rangle_{\hat{G}} = \int_G \hat{G}\,|\mathcal{A}\rangle\,\mathrm{d}\hat{G}, \tag{2}$$

where $\hat{G}$ is an element of the group G. This averaging can be performed formally over the Dirac ket, but is more conveniently carried out by choosing a convenient basis in which to write explicitly the feature vector. Furthermore, one should keep in mind that Haar integration – just as any averaging procedure – reduces the descriptive power of the representation. In other terms, structures that are distinct in terms of $|\mathcal{A}\rangle$ might be indistinguishable when represented in terms of $|\mathcal{A}\rangle_{\hat{G}}$. For example a Haar integration of $\langle \alpha \mathbf{r} | \mathcal{A} \rangle$ over the translations $\hat{t}$ yields a constant scalar that counts the number of atoms of type α that are present in the structure.[58] In order to avoid loss of resolving power, one can perform the average over tensor products of the atom density, *i.e.* evaluate the density at two different points and average over the simultaneous application of the symmetry operation to both points. To be concrete, let us derive explicitly this representation for a Gaussian smearing function $\mathcal{N}_{\sigma^2}(\mathbf{r}) = \exp(-\mathbf{r}^2/2\sigma^2)$. To retain structural information, we compute a translationally-symmetrized representation based on a two-point evaluation of the atom density:

$$\begin{aligned}
\langle \alpha \mathbf{r} \alpha' \mathbf{r}' | \mathcal{A}^{(2)} \rangle_{\hat{t}} &= \sum_{\substack{i \in \mathcal{A}, \alpha \\ j \in \mathcal{A}, \alpha'}} \int_{\mathbb{R}^3} \mathrm{d}\hat{t} \Big[\mathcal{N}_{\sigma^2}\left(\hat{t}\mathbf{r}' - \mathbf{r}_j\right) \\
&\qquad\qquad \mathcal{N}_{\sigma^2}\left(\hat{t}\mathbf{r} - \mathbf{r}_i\right) \Big] \\
&= \sum_{\substack{i \in \mathcal{A}, \alpha \\ j \in \mathcal{A}, \alpha'}} \int_{\mathbb{R}^3} \mathrm{d}\mathbf{t} \Big[\mathcal{N}_{\sigma^2}\left(\mathbf{r}' + \mathbf{t} - \mathbf{r}_j\right) \\
&\qquad\qquad \mathcal{N}_{\sigma^2}\left(\mathbf{r} + \mathbf{t} - \mathbf{r}_i\right) \Big] \\
&= \sum_{\substack{i \in \mathcal{A}, \alpha \\ j \in \mathcal{A}, \alpha'}} \mathcal{N}_{2\sigma^2}\left(\mathbf{r} - \mathbf{r}' - \mathbf{r}_i + \mathbf{r}_j\right) \\
\Rightarrow \langle \alpha \alpha' \mathbf{r} | \mathcal{A}^{(2)} \rangle_{\hat{t}} &= \sum_{\substack{i \in \mathcal{A}, \alpha \\ j \in \mathcal{A}, \alpha'}} \mathcal{N}_{2\sigma^2}\left(\mathbf{r} - \mathbf{r}_{ij}\right),
\end{aligned} \tag{3}$$

where $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$ and $\mathbf{r} - \mathbf{r}'$ has been replaced with $\mathbf{r}$. Note that translational averaging reduced by three the number of independent variables, and that the symmetrized density takes the structure of a many-body expansion of the potential energy truncated up to the pair contributions. In other words a symmetrized pair-density representation of an atomic structure can be decomposed into a sum of representations centered on each of the atoms. Moreover while one could consider all the pairs in the representation, the nearsightedness principle of electronic matter,[59] which underlies

most linear-scaling electronic structure methods,[60–63] and the clear computational advantage deriving from restricting the range of atomic pairs that need to be included in the sum, motivates the limitation of the atomic neighborhood to a sphere of radius r_c centered on each atom through a cutoff function $f_c(r)$ that is zero for $r > r_c$. To simplify notation, we then introduce an atom-centered symmetrized density representation

$$\langle \alpha \mathbf{r} | \mathcal{X}_i \rangle = \sum_{j \in \mathcal{X}_i, \alpha} \mathcal{N}_{2\sigma^2}(\mathbf{r} - \mathbf{r}_{ij}) f_c(r_{ij}), \tag{4}$$

where $\mathcal{X}_i$ is an atomic environment centered on atom i that includes all the neighbors within a sphere of radius r_c. The cutoff function should smoothly decay to zero to avoid introducing a discontinuity with respect to atoms entering/leaving the atomic neighborhood in the representation. Using this notation, one can write

$$\langle \alpha \alpha' \mathbf{r} | \mathcal{A}^{(2)} \rangle_{\hat{t}} = \sum_{i \in \mathcal{A}, \alpha'} \langle \alpha \mathbf{r} | \mathcal{X}_i \rangle \,. \tag{5}$$

The environment-centered features $|\mathcal{X}_i\rangle$ are not rotationally invariant, and so one can proceed to the symmetrization over the rotation group. Using the z-y-z Euler parametrization, one can compute

$$\begin{aligned}
\langle \alpha \mathbf{r} | \mathcal{X}_i^{(1)} \rangle_{\hat{R}} &= \sum_{j \in \mathcal{X}_i, \alpha} f_c(r_{ij}) \int_{\mathrm{SO(3)}} \mathcal{N}_{2\sigma^2}(\hat{R}\mathbf{r} - \mathbf{r}_{ij}) \mathrm{d}\hat{R} \\
&= 2\pi \sum_{j \in \mathcal{X}_i, \alpha} f_c(r_{ij}) \int_0^{\pi} \sin\theta \mathrm{d}\theta \int_0^{2\pi} \mathrm{d}\phi \\
&\qquad \exp\left[-\frac{r^2 + r_{ij}^2 - 2 r r_{ij} \cos\theta}{4\sigma^2} \right] \\
&= 8\pi^2 \sum_{j \in \mathcal{X}_i, \alpha} f_c(r_{ij}) \sinh[r r_{ij}/2\sigma^2] \\
&\qquad (r r_{ij}/2\sigma^2)^{-1} \exp\left[-(r^2 + r_{ij}^2)/4\sigma^2 \right] \\
\langle \alpha r | \mathcal{X}_i^{(1)} \rangle_{\hat{R}} &\approx \sum_{j \in \mathcal{X}_i, \alpha} f_c(r_{ij}) r_{ij}^{-1} \mathcal{N}_{2\sigma^2}(r - r_{ij}),
\end{aligned} \tag{6}$$

where we note that the integration makes the orientation of $\mathbf{r}$ irrelevant, and we write the feature vector as a function of $r = \| \mathbf{r} \|$. Some constant factors and the $\mathcal{N}_{2\sigma^2}(r + r_{ij})$ term have been omitted because they do not contribute to the representation since $r, r_{ij} > 0$ with σ is relatively small. Note also that we have introduced in the definition of $\langle \alpha r | \mathcal{X}_i^{(1)} \rangle_{\hat{R}}$ an additional factor of r, so that

$$\begin{aligned}
\int_{\mathbb{R}^3} \langle \mathcal{X}_i^{(1)} | \alpha \mathbf{r} \rangle_{\hat{R}} \langle \alpha \mathbf{r} | \mathcal{X}_j^{(1)} \rangle_{\hat{R}} \mathrm{d}\mathbf{r} = \\
\int_0^{\infty} \langle \mathcal{X}_i^{(1)} | \alpha r \rangle_{\hat{R}} \langle \alpha r | \mathcal{X}_j^{(1)} \rangle_{\hat{R}} \mathrm{d}r.
\end{aligned} \tag{7}$$

This symmetrized density $\langle \alpha r | \mathcal{X}_i^{(1)} \rangle_{\hat{R}}$ is essentially a 2-body correlation function resulting from a Gaussian kernel density estimation (KDE). The body order naturally characterizes the amount of information included in an invariant density representation.

It is clear that this procedure is very general, and can be applied to any tensor power of the density, both when integrating over translations and when integrating over rotations. Increasing the order μ of the product in the integration over $\hat{t}$ leads to $\mu - 1$ nested sums over the atomic neighborhood which might not be computationally favorable. As illustrated in Fig. 1 for the case of $\nu = 2$, increasing the order ν of the tensor product in the integral over the continuous rotation group similarly increases the body order of the structural correlations described by $|\mathcal{X}_i^{(\nu)}\rangle_{\hat{R}}$. If one did so while writing explicitly the environment ket $|\mathcal{X}_i\rangle$ as a sum over neighbors, this procedure would increase the order of the sum over neighboring atoms. One can, however, also proceed by expanding $|\mathcal{X}_i\rangle$ in an appropriate basis, *e.g.* a combination of radial functions $R_n(r)$ and spherical harmonics

$$\langle \alpha n l m | \mathcal{X}_i \rangle = \int \mathrm{d}\mathbf{r}\, R_n(r) Y_l^m(\hat{\mathbf{r}}) \langle \mathbf{r} | \mathcal{X}_i \rangle \,, \tag{8}$$

in which case higher-order invariants can be written as sums over the expansion coefficients,

$$\begin{aligned}
\langle \alpha n \alpha' n' l | \mathcal{X}_i^{(2)} \rangle_{\hat{R}} = \frac{1}{\sqrt{2l+1}} \sum_m (-1)^m \langle \alpha' n' l m | \mathcal{X}_i \rangle \\
\langle \alpha n l - m | \mathcal{X}_i \rangle \,.
\end{aligned} \tag{9}$$

The flexibility of this framework allows links to be drawn between several representations that might otherwise look quite dissimilar. The type of smearing function used to construct the atomic density, the basis onto which the density is represented (real space grid, orthonormal basis set, *etc.*), can impact the effectiveness and the computational efficiency of the resulting implementation but do not change the fundamental nature of the invariant representation. For example, the choice of Gaussian smearing and a basis of radial functions corresponds to the smooth overlap of atomic positions (SOAP) framework,[14,46] with the power spectrum and the bispectrum corresponding to rotational averages with $\nu = 2$ and $\nu = 3$ respectively. The computation of these coefficients involves the evaluation of several costly special functions.[53] Even if the cost of evaluating SOAP features can be reduced greatly by the introduction of approximations and numerical workarounds,[64] the use of both a smooth atom density and a smooth basis set might seem redundant and costly. This led Drautz[47] to use Dirac distributions in the representation of the density, and obtain smoothness by truncating the basis set on which this density is expanded. The resulting invariant representations correspond precisely to the $g \to \delta$ limit of the SOAP power spectrum, bispectrum and higher-ν invariants, but can be expressed in terms of simpler mathematical functions.

The expansion on a complete basis set of the atomic density ensures the general applicability of a representation but it also increases its computational cost by probing regions of the configurational space that are not relevant for a given system. The symmetry functions[40] framework make it possible to use the knowledge of the system at hand to carefully tailor a representation of the atomic environment. The resulting representation can be interpreted as a projection on these symmetry functions fixed in particular regions of the configurational space with the δ-limit of the $(\nu + 1)$-body invariant ket,

$$\langle \alpha G_2 | \mathcal{X}_i \rangle = \int \mathrm{d}r\, G_2(r) \langle \alpha r | \mathcal{X}_i^{(1)} \rangle_{\hat{R}, g \to \delta} \,, \tag{10}$$

where $\langle \alpha r | \mathcal{X}_i^{(1)} \rangle_{\hat{R}, g \to \delta} = \sum_{j \in \mathcal{X}_i, \alpha} \delta(r - r_{ij}) f_c(r_{ij})$ and δ is the Dirac distribution. Other recently introduced feature vectors for atomistic learning, such as the FCHL[44] and the MBTR[48] representations, use an adaptive basis to smooth the δ-limit of the $(\nu + 1)$-body invariant ket $|\mathcal{X}_i^{(\nu)}\rangle_{\hat{R}}$, effectively constructing a kernel

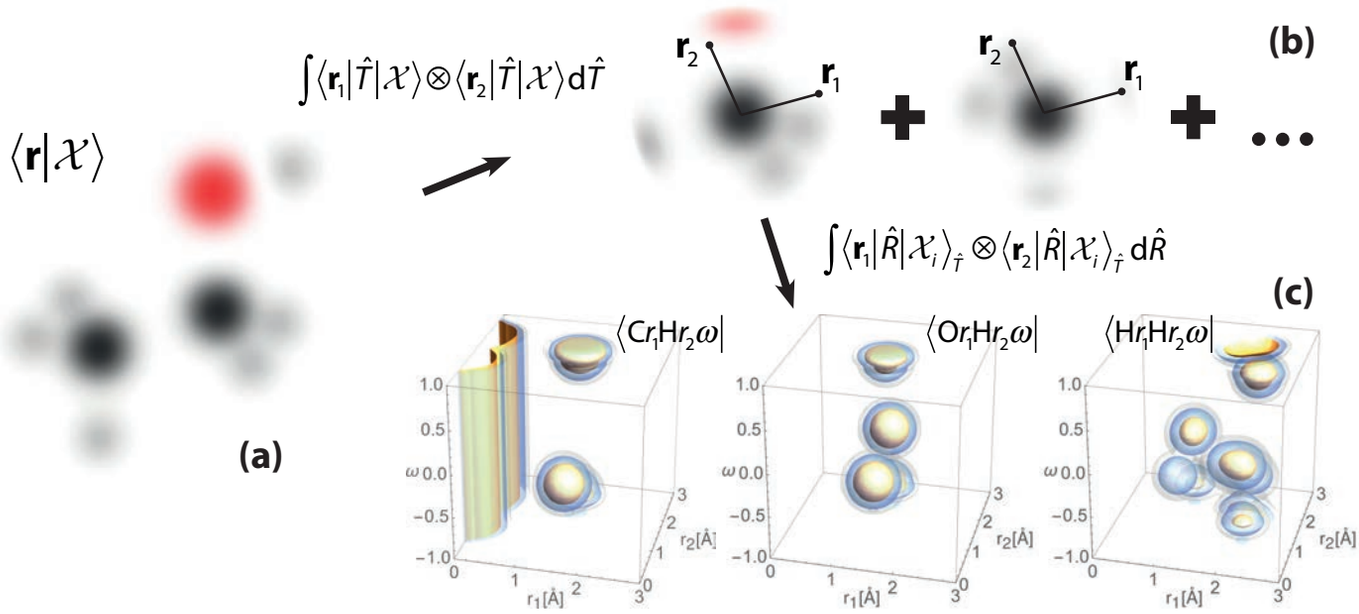

Fig. 1. A graphical summary of the steps leading from a decorated atomic density to the 3-body invariant representation of ethanol in real space $|\mathcal{X}_i^{(2)}\rangle_{\hat{R}}$. (a) The geometry of a small molecule is mapped into a smooth atom density using a Gaussian smearing function. The chemical composition represented by the elemental ket $|\alpha\rangle$ is color coded: carbons are black, oxygen is red and hydrogens are grey. (b) The symmetrization over the translational group of a two-point density results in the decomposition of the representation into a sum of atom-centered contributions where a finite cutoff has been applied (see Eqn. (4)). (c) The symmetrization over the rotational group with $\nu = 2$ delivers the 3-body invariant representation. Some isocontours of $\langle \alpha r_1 \beta r_2 \omega | \mathcal{X}^{(2)} \rangle_{\hat{R}} / r_1 r_2$ associated with the central carbon atom illustrate some of the real space features extracted with the atom density framework. Adapted from ref. [58].

density estimate of the invariant correlation function. These two representations differ by the choice of kernel functions and how they encode the chemical information, with the FHCL features using a kernel to encode the similarity between different elements, similarly to what was done in ref. [14].

The density-based representation framework makes it possible to rigorously formulate a hierarchy of invariant representations and shows that several commonly used descriptions of atomic structures and environments actually contain a similar amount of information. This formal connection is also reflected in several recent extensive empirical benchmarks,[65–67] that show that many of these representations actually perform similarly in terms of model accuracy, while the main difference between them is their computational cost.

It is also worth mentioning that density-based invariants can be generalized to yield feature vectors of the form $\left|\mathcal{X}^{(\nu)}\lambda\mu\right\rangle_{\hat{R}}$ that transform covariantly under rotations of the reference frame as the spherical harmonics Y_λ^μ,[53,54] providing a symmetry-adapted basis to learn properties such as atomic forces, elastic moduli or dielectric response tensors, which also rotate rigidly under $\mathcal{SO}(3)$ group operations. Using a generic atom-centered symmetry-adapted representation has proven to be more effective[68] than frameworks that assume a rigid molecular frame to achieve covariance of the predicted properties,[69,70] and has made it possible to learn an atom-centered decomposition of a scalar field like the electron density.[23,71] Although learning schemes based on covariant features or kernels[55,72–74] could in principle be used to machine-learn directly the inter-atomic forces rather than the underlying atomic potential, enforcing energy conservation has proven difficult. For this reason, most of the existing machine-learning interatomic potentials are built to predict the potential, although they can incorporate forces as an indirect learning target.[20,40,75–77]

3. Machine Learning Quantum Mechanics

ML algorithms for regression[78] aim to construct a model $y = F(\mathcal{A})$ that can predict accurately the properties of a structure. The internal parameters of the model are determined by optimizing the accuracy of prediction over a set of training structures, $\{\mathcal{A}_i, y_i\}$, and their accuracy with respect to that reference can be improved systematically by increasing the size of the training set.[79] One of the early applications of ML to the prediction of atomic-scale properties aimed at obtaining an accurate model of the potential energy surface (PES), which is crucial to assess the stability of a given configuration, and whose sampling underlies the evaluation of the thermodynamic properties of a system.[80] Contrary to traditional FFs, which assume physics-inspired functional forms for the interactions, and often use experimental observable as fitting targets, ML interatomic potentials (MLIPs) don't assume a fixed functional form, and usually rely on electronic-structure calculations as a reference. In many cases, this more general, data-driven approach has been shown to result in more transferable and accurate models.[19,20,41,81] Besides the PES, ML models have also been successful at predicting other zero Kelvin properties such as chemical shieldings, band gaps, electron affinities, electron transfer integrals and static isotropic polarizabilities.[14,15,21,77,82–85] While considerable success has also been shown in using ML to predict complex properties that cannot be seen as arising from an individual atomic configuration (*e.g.* the free-energy of a state, the toxicity or pharmaceutical activity of a molecule, *etc.*), here we will focus entirely on the well-defined task of building a surrogate quantum model, which can sidestep the solution of the Schrödinger equation and predict the properties of a specific atomic configuration. In this section we summarize the regression methods that have been applied to perform such prediction. While the main focus will be on the construction of interatomic potentials, we will keep the discussion as general as possible, and mention how the different approaches should be modified to deal with other classes of properties.

A scalar property $y(\{\mathbf{r}_i, \alpha_i\})$ of a system $\mathcal{A}$ of N atoms of species α_i, located at positions $\mathbf{r}_i$, can be expressed formally as a function of an abstract vector of features $|\mathcal{A}\rangle$ that represents the structure,

$$y(\mathcal{A}) = F(|\mathcal{A}\rangle). \tag{11}$$

The problem of modeling $y(\mathcal{A})$ can therefore be decomposed into the problem of providing a concrete formulation of the feature vector (that we have discussed in detail in the previous Section) and that of determining the functional form of the approximating model *F*. Irrespective of the regression technique used, most of the transferable property models that have been introduced in recent years decompose a property associated to a set of atoms $\mathcal{A}$ into atom-centered contributions, *i.e.*

$$y(\mathcal{A}) = \sum_{i\in\mathcal{A}} f(|\mathcal{X}_i\rangle), \tag{12}$$

where *f* is a trained ML model and $\mathcal{X}_i$ indicates the atomic environment centered on atom *i* of structure $\mathcal{A}$. This choice can be motivated as a consequence of imposing the invariance of the property on the absolute position of the system (see Eqn. (3)), and – together with the limitation of the range of each environment to a region centered on the *i*-th atom – yields models of great transferability, since it allows breaking down the properties of large, complex configurations into a sum of contributions that only depend on the position of a few dozen atoms. In the cases in which this ansatz is not justified (*e.g.* for properties such as ligand binding affinity, or in the presence of significant long-range interactions) other strategies for combining local environments predictions like the REMatch kernel should be considered.[84]

Linear models based on permutation invariant polynomials (PIPs) have been very effective at reproducing accurately chemical reactions between small molecules[27,41,42] and to build efficient MLIPs with the many-body tensor (MBT) framework[43] that extends them to more complex systems.[86,87] Similarly linear models based on the *n*-body correlation function[47,50,88–90] have shown great promise. Fully non-linear models based on artificial neural networks (ANN) have, however, been the most popular this far. ANNs have been constructed based on the the expansion of the radial (and angular) distribution function on a basis such as the Behler-Parrinello symmetry functions,[81,91–97] Zernike polynomials,[98] Chebychev polynomials,[99] Gaussians,[77,100–102] and proved very successful at investigating the properties of complex systems.[85,103–108] Another class of models that have been both very popular and successful is based on Gaussian process regression (GPR),[109] that is formally equivalent to kernel ridge regression (KRR) and can be seen as a middle-ground solution that introduces non-linearity in the form of a kernel function $k(\mathcal{X}, \mathcal{X}')$ built on pairs of feature vectors, but effectively translates into a linear regression problem that uses (some of) the training set structures as the basis on which the structure–property relation is constructed. GPR has been used to predict the stability of molecules and solids[14,15,19,20,83,84,110–114] and build MLIPs for elemental solids,[115–118] nano clusters,[119] isolated molecules[76] and molecular liquids[120] as well as for the direct prediction of other quantum mechanical properties.[15,21,68,82,121–123]

In the most straightforward form, a GPR model built on a kernel function *k* can be written based on a set of *N* training structures $\{\mathcal{T}_n\}$, and the associated properties y_n. Assuming a Gaussian likelihood, and an additive, atom-centered property model, the prediction for a structure $\mathcal{A}$ becomes

$$y(\mathcal{A}) = \sum_{n=1}^{N} x_n K_{\mathcal{A}\mathcal{T}_n}, \tag{13}$$

where $K_{\mathcal{A}\mathcal{T}_n} = \sum_{i\in\mathcal{A}}\sum_{j\in\mathcal{T}_n} k(\mathcal{X}_i, \mathcal{T}_{n,j})$ and the kernel function k(·, ·) quantifies the similarity between the local environments of $\mathcal{T}_n$ and the centered structure $\mathcal{X}_i$. The key ingredient of this model is the kernel function that – subject to a few conditions such as positive definiteness – defines an inner product between the inputs $k(\mathcal{X}_i, \mathcal{X}_j)$. The representer theorem[124] guarantees that the kernel can be associated with an inner product between vectors in a Hilbert space, *i.e.* $k(\mathcal{X}_i, \mathcal{X}_j) = \langle\mathcal{X}_i|\mathcal{X}_j\rangle$. The use of the Dirac notation underlines the independence from the basis, *i.e.* representation or features, used to effectively quantify the similarity between atomic configurations. In some cases – for instance in the case of the SOAP representation discussed in the previous Section – it may be possible to write explicitly the feature vectors associated with a given kernel.

GPR is often preferred over the more sophisticated non-linear models because of its ease of use: it has a single interpretable hyperparameter σ, and the solution for the weights x_n has the closed form

$$\boldsymbol{x} = \boldsymbol{K}^{-1}\boldsymbol{y}, \tag{14}$$

where $K_{nm} = K_{\mathcal{T}_n\mathcal{T}_m} + \sigma^2\delta_{nm}$ is the kernel matrix between the *N* training inputs and y_n is the property associated with structure $\mathcal{T}_n$; σ corresponds to an expected Gaussian noise in the references $\boldsymbol{y}$ so it can account for small discrepancies in the convergence of the electronic structure method that are often found across a training set. In the language of kernel ridge regression, Eqn. (14) can be obtained by minimizing the loss

$$\mathcal{L}(\boldsymbol{x}) = \sum_n |y(\mathcal{T}_n) - y_n|^2 + \sigma^2\sum_n x_n^2. \tag{15}$$

It should be mentioned that GPR provides a simple approach to compute derivatives of the target properties with respect to atomic positions, *e.g.* the force consistent with the model, in which case *y* represents the PES of a configuration. Such derivatives are easily expressed in terms of derivatives of the kernel, *i.e.*

$$\frac{\partial y(\mathcal{X}_i)}{\partial \mathbf{r}_i} = \sum_{n=1}^{N}\sum_{j\in\mathcal{T}_n} x_n \frac{\partial}{\partial \mathbf{r}_i} k(\mathcal{X}_i, \mathcal{T}_{n,j}). \tag{16}$$

Derivatives can also be incorporated in the learning procedure,[75,76,109,114,123,125] by including the discrepancy between reference and predicted values in the loss Eqn. (15). Building a symmetry-adapted GPR model for properties that have a tensorial nature requires the construction of covariant kernels,[54,55] that describe the correlations between the spherically-covariant components of the target property,

$$y_\lambda^\mu(\mathcal{A}) = \sum_{n=1}^{N}\sum_{m=-\lambda}^{\lambda} x_{nm}\,[K_{\mathcal{A}\mathcal{T}_n}]_{m\mu}^{\lambda}\,. \tag{17}$$

For instance, a kernel which fulfills these symmetry requirements can be constructed based on λ-SOAP features,[54]

$$k_{\mu\mu'}^{\lambda}(\mathcal{X}, \mathcal{X}') = \sum_{nn'll'} \left\langle \mathcal{X}^{(2)}\lambda\mu \middle| nn'll' \right\rangle \left\langle nn'll' \middle| \mathcal{X}'^{(2)}\lambda\mu \right\rangle. \tag{18}$$

Finally, the probabilistic nature of GPR also allows one to estimate the uncertainty associated with the prediction

$$\sigma_y(\mathcal{A}) = \sigma^2 + K_{\mathcal{A}\mathcal{A}} - \boldsymbol{K}_{N\mathcal{A}}^T\boldsymbol{K}^{-1}\boldsymbol{K}_{N\mathcal{A}}. \tag{19}$$

The drawback for such simplicity is the computational cost associated with the training phase – which scales cubically with the training set size – and the need to use the full training set as a basis to perform predictions. To address this issue, many approximations of the exact kernel matrix have been proposed,[126,127] among which the projected process (PP) approximation[126,128] has been shown to be quite practical to include force references[75,125] and effective from the point of view of the cost and accuracy of predictions.[118,129] The PP method introduces M pseudo inputs to approximate the GP prior which practically reduces the cost of training to the inversion of a $M \times M$ matrix, and ensures that predictions only require computing kernels between the new configurations and the M pseudo inputs:

$$\begin{aligned} y^{\mathrm{PP}}(\mathcal{A}) &= \boldsymbol{K}_{M\mathcal{A}}^{T}\tilde{\boldsymbol{K}}^{-1}\boldsymbol{K}_{MN}\boldsymbol{y}, \\ \sigma_{y}^{\mathrm{PP}}(\mathcal{A}) &= \sigma^{2} + K_{\mathcal{A}\mathcal{A}} - \boldsymbol{K}_{M\mathcal{A}}^{T}\boldsymbol{K}_{MM}^{-1}\boldsymbol{K}_{M\mathcal{A}} \\ &\quad + \boldsymbol{K}_{M\mathcal{A}}^{T}\tilde{\boldsymbol{K}}^{-1}\boldsymbol{K}_{M\mathcal{A}}, \end{aligned} \tag{20}$$

where $\tilde{\boldsymbol{K}} = \boldsymbol{K}_{MM} + \sigma^{-2}\boldsymbol{K}_{\mathrm{NM}}^{\mathrm{T}}\boldsymbol{K}_{\mathrm{NM}}$, $\boldsymbol{K}_{\mathrm{MM}}$ indicates the kernel matrix between pseudo inputs and $\boldsymbol{K}_{\mathrm{NM}}$ the matrix between training points and pseudo inputs. For simplicity, the pseudo inputs (or active points) can be chosen directly from the training set and they represent a new basis in which the regression is performed. To maximize the cost reduction and the accuracy of the model, one needs to sample the active set carefully. Selecting randomly the active inputs is far from optimal so several approaches have been proposed[128,130–132] among which Farthest Point Sampling (FPS),[133] a greedy method that maximizes diversity, or a CUR decomposition[125,134] of the feature matrix associated with the training set, which minimizes the effect of the PP on the kernel matrix, have allowed significant reductions of the computational cost with minimal degradation of the accuracy.[118,129]

ML algorithms include recipes to train their parameters, e.g. Eqn. (14), but they do not specify how to determine hyperparameters such as the regularization σ for GPR, the number of layers in an ANN and the cutoff radius r_c in the power spectrum representation, which can influence heavily the quality of the model. In the Bayesian context these hyperparameters can be interpreted as priors that should be inferred from our knowledge of the physical system,[125] or thought of as parameters that need to be optimized. In principle the best parameters should allow for the lowest possible prediction error on all possible inputs. Given that one can only work on a finite-sized set of references, the problem becomes to find the parameters that best reproduce the available references and at the same time generalize well to unknown inputs. The performance of a model is measured by comparing the predicted values and the reference values with metrics such as the mean absolute error (MAE) or the root mean square error (RMSE). An effective technique to avoid overfitting these parameters, *i.e.* specialize the model for the training set which leads to poor generalization performances, is the so called k-fold cross-validation where the performances are evaluated on several subsets of the training set (see Hansen *et al.*[111] for more details). Cross validated scores are more likely to match the generalization error which is a good basis to rank models and determine the optimal set of hyperparameters.[135] Learning curves are another standard diagnostic tool to characterize the performance of ML models. From statistical theory, the error of a given model decreases as a power-law with the size of the training set.[79] Fig. 2 shows, on a logarithmic scale, three learning curves for models trained on datasets of molecular crystal polymorphs to reproduce their lattice energies. The GPR model performances vary with the considered training set because the learning rates (slopes of the curves) and off-sets are different. These curves are very useful because they help differentiate between models that have a small offset and learning rates with models that have a larger off-set but also steeper slopes (see Fig. 4 for an example). Indeed, building a 'good' model with as few references as possible might be favored over a model that has a better learning power but poorer performances with few samples.

Even though learning curves and cross-validation procedures can benchmark quantitatively the ability of a model to perform well in production, demonstrating the performance of a model on practical test cases is typically more compelling. For example, Fig. 3 shows how the ShiftML model for the ^{1}H chemical shifts[139] is able to identify the crystal structure observed experimentally with NMR spectroscopy of two molecular materials as well as GIPAW DFT, the reference method used to train it. In the better-established case of the construction of MLIPs, several recent works have started to compare systematically the ability of different schemes to reproduce *ab initio* energies and forces,[140] the short range interaction in the MB-pol water model,[65] the vibrational spectra of H_2CO,[141] the radial and angular distribution functions of copper and silica and the equation of state of three binary alloys.[67] Overall these studies show that all of the models considered were able to reproduce observables within the expected accuracy of the underlying electronic structure reference. In light of the substantially equivalent asymptotic accuracy of different approaches, the preference for one MLIP over another depends more on practical considerations such as training data efficiency, computational cost, simplicity of use *etc*.

The ability of a ML model to reproduce the results of reference calculations on a validation/test set makes it possible to assess its overall quality, but it does not guarantee that the predictions are equally accurate. A reliable uncertainty estimate that provides an assessment of the model accuracy *for a specific prediction* is key to allow for a wider community of researchers to rely on ML models. A punctual quantification of ML uncertainty is also useful as a criterion for the iterative improvement of a model's training set with active learning,[142–145] as one would like to incorporate additional reference data in the regions that correspond to the least accurate predictions. Several techniques such as GPR, Bayesian neural networks (BNN)[146,147] and ensemble models[148] have been developed to provide an estimate of the uncertainty associated with predictions. Model ensembles, which estimate uncertainty by performing multiple

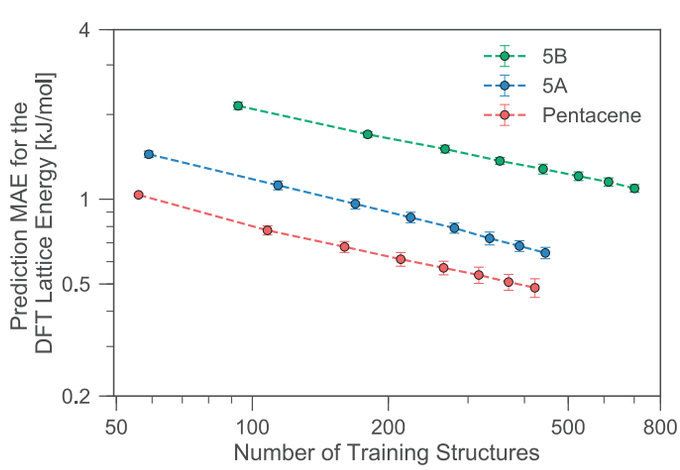

Fig. 2. Learning curves for the lattice energy predictions of pentacene, 5A and 5B datasets, plotted on on a logarithmic scale. For each training sample size, models are built several times on random subsets of the full training set and predictions are made on a fixed-size random subset of the training set. The test MAE and error bars are, respectively, average and standard deviation over the random subset predictions. All hyper-parameters of our ML model are fixed except for the regularization parameter σ in the GPR model which is optimized on the fly at each training. Adapted from ref. [15].

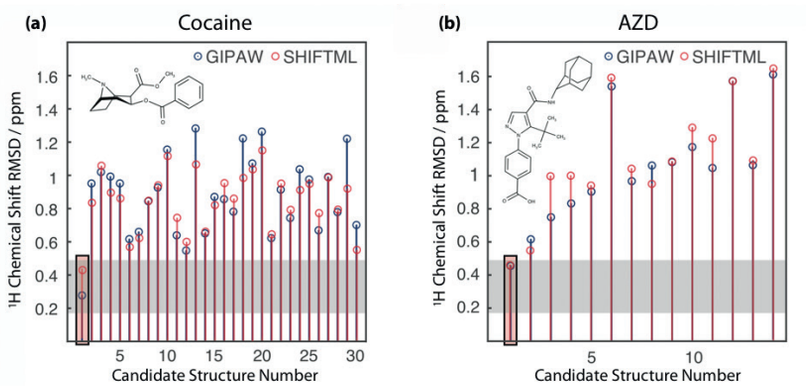

Fig. 3. Structure determination for cocaine (a) and AZD8329 (b) obtained by comparing calculated and experimental ^{1}H chemical shifts for the most stable structures obtained with CSP. The total RMSE between experimentally measured shifts (NMR spectroscopy) and shifts calculated with GIPAW[136,137] (blue) and ShiftML[21] (red) is shown for every hypothetical structure. The shaded area represents an estimation of the confidence intervals for the total RMSE computed with GIPAW. The candidates that have RMSEs within this range are the most likely observed crystal structures using a chemical shift-based solid-state NMR crystallography protocol.[138] Adapted from ref. [21].

predictions for each input, have been quite popular[129,142,149,150] because of their simplicity and flexibility. The resampling approach in particular[151–154] is based on the training of a family of models, N_r different subsets of the training data. It produces a non-parametric estimate of the predictions distribution $P(y|\mathcal{A})$ whose moments are given by

$$\bar{y}(\mathcal{A}) = \frac{1}{N_r}\sum_i y^{(i)}(\mathcal{A})$$
$$\bar{\sigma}^2(\mathcal{A}) = \frac{1}{N_r - 1}\sum_i [y^{(i)}(\mathcal{A}) - \bar{y}(\mathcal{A})]^2, \quad (21)$$

where $y^{(i)}(\mathcal{A})$ is the prediction of the i^{th} resampled model.

While the training cost is increased N_r times, uncertainty predictions come with the estimate of $y(\mathcal{A})$ at essentially no extra cost for GPR – which is typically dominated by the evaluation of the kernel. In the case of ANN, an ensemble of models provides a practical way of estimating the uncertainty, although in this case the overhead can be significant, and linear in N_r. To avoid such overhead, as well as the increased training cost, dedicated schemes that avoid training multiple models have been developed specifically for ANN.[155–157]

4. Optimizing the Representations

As discussed previously, representations of an atomic structure for atomic scale simulations should provide a concise but complete description of its structure and composition. Ensuring that these features follow the basic symmetries of the target property is an essential condition, but does not guarantee optimal performance of the resulting model. One way to optimize a model for a given regression task is to consider multiple kinds of representations and build a weighted combination, and treat the weights as hyperparameters. This line of reasoning has been used to optimize the performance of a ML scheme to estimate the formation energy of small molecules[84] and the chemical shieldings in molecular crystals.[21] Both applications compound local descriptions with increasing cut-off spheres and decreasing weights, outperforming the best individual representation model. The decaying weights assigned to representations with larger cutoffs reflect the multi-scale nature of the interactions that affect the values of chemical shieldings and of the molecular cohesive energy, which are often determined predominantly by the closest neighboring atoms and depend less markedly on atoms that are farther away. To confirm this intuition Willatt *et al.*[52] compare a similar mixture of representation with radially scaled representations to model the formation energy of small molecules which corresponds to Eqn. (4) with

$$\langle \alpha \mathbf{r}|\mathcal{X}_i\rangle = \sum_{j\in\mathcal{X}_i,\alpha} \mathcal{N}_{2\sigma^2}(\mathbf{r}-\mathbf{r}_{ij}) f_c(r_{ij}) u(r_{ij}), \quad (22)$$

where $u(r_{ij})$ is a flexible radial scaling that reduces the weight of atoms in the far field. The prediction accuracies of the mixture of representations and radial scaling model are shown to be very similar after independent optimization of the model parameters with cross-validation. This example showcases how incorporating physical insights about the target property into the representation helps in building more effective models.

Another scheme by which the atom-density framework can be generalized to reflect structure–property relations builds upon the similarities in the behavior of different chemical elements, which is reflected in the well-known trends observed along the periodic table. Discarding such knowledge by considering each chemical species as completely different seems wasteful and even impractical when working on large subsets of the periodic table.[66,99,158] Following this intuition, De *et al.*[14] formulated an 'alchemical' kernel to supplement the SOAP power spectrum with the correlations between chemical species based on Pauling electronegativity. With the same mindset a distance across the periodic table has been proposed to learn properties across chemical composition space in the FCHL representation.[44]

Rather than using elemental properties to define *a priori* the similarity between elements, the optimization of the representation of chemical space can be set as an additional objective of the ML algorithm. Then, the chemical features that characterize each element are learnt directly according to the dataset and target property at hand. Several applications of ANN to model the PES of molecules or solids use the stoichiometry as an input and the resulting features tend to match well with the structure of the periodic table.[96,100,159] An alternating least square optimization procedure has been proposed to achieve similar results within the GPR framework.[52] It effectively corresponds to finding the best projection of the abstract elemental kets $|\alpha\rangle$ on an 'elemental feature' basis $|J\rangle$, *i.e.* an embedding space, of dimension d_J by optimizing the coefficients $u_{\alpha J} = \langle J|\alpha\rangle$ within the modified power spectrum representation

$$\sum_{\alpha\alpha'} u_{\alpha J} u_{\alpha' J'} \sum_m (-1)^m \langle \alpha n l m|\mathcal{X}_i\rangle \langle \alpha' n' l - m|\mathcal{X}_i\rangle . \quad (23)$$

In Fig. 4, the learning curves of several chemically compressed models are compared with a baseline model, a compound model and the model taken from ref. [44] for a chemically diverse benchmark dataset. The compressed models tend to saturate because

the low-dimensional 'elemental features' are not sufficiently descriptive to account for the differences between the 39 elements in the dataset. Nevertheless, in the limit of small training set size the compressed models (with d_J = 2, 3, 4) clearly outperform the baseline model. The compound model (grey line) that combines both the baseline representation and the representation with d_J = 4 avoids the saturation of the learning and retains the improved learning for small training set sizes.

After optimization, the embedding space contains information on the similarity between elements with respect to the target property. The 'elemental features' obtained on a dataset of perovskites (62 different elements) and a model trained to predict their formation energy is shown in Fig. 5 for d_J = 2. The resulting projection of the chemical elements evokes their positions in the periodic table which is highlighted by the coloring according to the periodic table group. Moreover, the spatial arrangement of the two-dimensional projection appears well correlated with the electronegativity. Such data-driven techniques are powerful since they adapt to the system and target property but this also comes at an increased computational cost. Furthermore, the optimized chemical space might not be transferable across classes of systems, or for the learning of different properties.

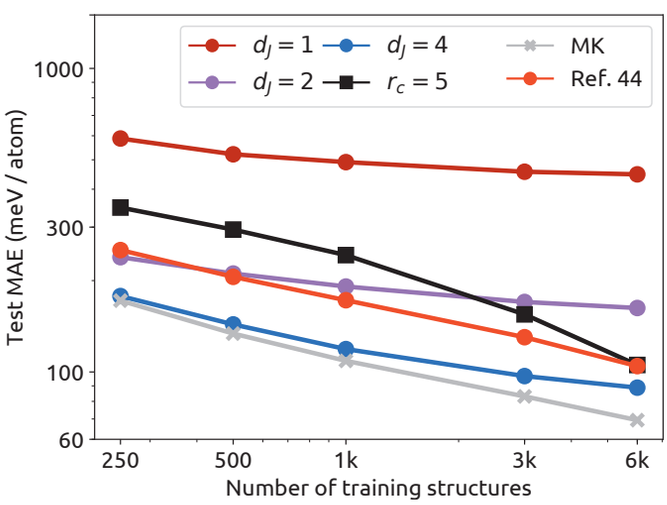

Fig. 4. Learning curves for the formation energy of the elpasolite crystals database using GPR.[112] The standard power spectrum curve is shown in black, the best curve from ref. [44] is shown in bright red and the optimized curves are shown in dark red (d_J = 1), purple (d_J = 2) and blue (d_J = 4). For each of these models, the kernel was constructed with r_c = 5 Å, n_{max} = 12 radial basis functions and l_{max} = 9 non-degenerate spherical harmonics. The compound model (shown in grey) combines three standard power spectrum representations and one chemically compressed representation (d_J = 4, r_c = 5) in the ratio 4 : 3 : 1 : 220. Adapted from ref. [52].

Besides data efficiency and the accuracy of predictions, numerical efficiency is also an essential criterion for a representation, since it affects the length and time scales of problems that it can be used with. For example Caro[64] proposed an approximation to compute the SOAP power spectrum which results in a clear speedup with a marginal loss of accuracy. Similarly, the FCHL representation has been reformulated[123] using much simpler functional forms increasing the numerical efficiency without impacting much the accuracy of the model. In addition to improve the cost of computing the feature vectors associated with a given representation, computational effort can also be cut by reducing the number of features that need to be computed and used as input of the ML model. ML schemes such as the CUR decomposition and the FPS scheme have been used to select a subset of the components of the power spectrum representation, and identify the most important parameters for Behler-Parrinello symmetry functions, obtaining simpler and more efficient models that were equivalent in performance to the full models.[134]

In closing, let us note that the Dirac notation that we have used to introduce the symmetrized atom-density framework also makes it possible to formulate many existing optimizations as the application of a linear operator that preserves the symmetries of the representation.[58] For example a rotationally invariant operator that acts on the chemical part of a representation has matrix elements

$$\langle \alpha n l m | \hat{U} | \alpha' n' l' m' \rangle = \delta_{nn'} \delta_{ll'} \delta_{mm'} \langle \alpha | \hat{U} | \alpha' \rangle \,, \qquad (24)$$

when written in the same basis of radial functions and spherical harmonics used in Eqn. (8). A low-rank expansion of such operator can be written as

$$\hat{U} \approx \sum_{J\alpha} u_{J\alpha} \, |J\rangle\langle\alpha| \,, \qquad (25)$$

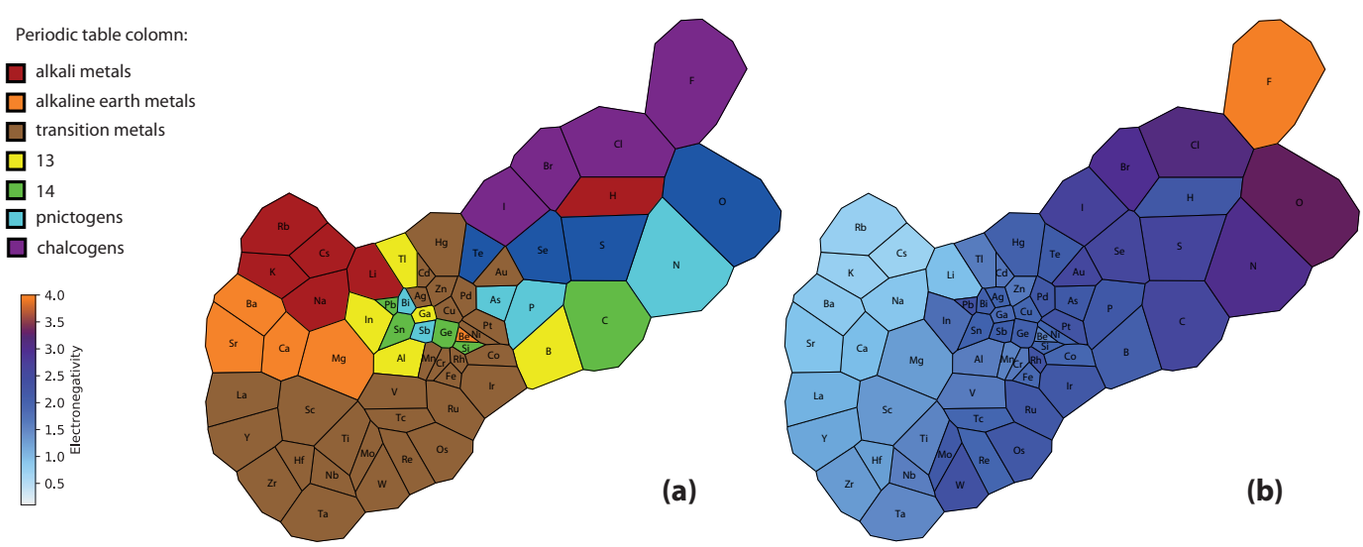

Fig. 5. Representation of the chemical space obtained by optimizing a model of the lattice energy for a set of perovskites[160] whose composition is based on combinations of 64 elements. The map corresponds to the coefficients $u_{\alpha J}$ with d_J = 2 (see Eqn. (23)). Each element is represented with a Voronoi cell where the facets are at the midpoints between neighboring elements. The elements are color coded according to (a) their group in the periodic table and (b) Pauling electronegativity.

corresponding to a transformation of the chemical space into low-dimensional basis $|J\rangle$. Applying such an operator to the power spectrum representation leads directly to Eqn. (23). Similar operators can be formulated resulting in the geometrical scaling introduced by Faber *et al.*[44] or in the radial scaling of Eqn. (22). The atom density-based framework is helpful to rationalize *ad hoc* attempts to improve representations, generalize them and formulate rigorously more complex modifications that would, for instance, couple the geometry and the composition channels.

5. Conclusion

In the last decade ML techniques have demonstrated their utility in the context of atomistic simulations, both by automating the post-processing of large amounts of data, *e.g.* molecular dynamics trajectories, and by improving the efficiency and/or the accuracy of the prediction of atomic scale properties – most notably through the construction of machine-learned interatomic potentials. Some consensus is starting to develop around the features of an effective ML model, *e.g.* the utility of incorporating symmetries and physical principles in the construction of representations of atomic neighborhoods, the importance of active-learning strategies and uncertainty quantification, and the need to balance computational efficiency, data efficiency and transferability. Moreover, several software packages for building MLIPs and predicting atomic-scale properties have been made publicly available, and have been interfaced with efficient simulation codes, helping their dissemination beyond the groups that focus on methodological developments.[98,125,161–165] ML models have already been integrated in a few well-established workflows of such as global structure optimization[166] and crystal structure prediction.[167] These applications, which need to generate and then screen thousands of configurations for a given system, and that do not need the models to be transferable across completely different compounds to yield a substantial acceleration, are the first candidates to benefit from ML. Methods that require sampling of fluctuations, such as molecular dynamics, are also very well suited to ML, as an effective interpolation scheme can cut down dramatically the cost needed to obtain accurate estimates of thermodynamic and time-dependent properties.[168,169] One of the main challenges to make the use of ML more stream-lined, and to obtain more transferable models, lies in efficiently generating a reference dataset, which requires the sampling of a large variety of configurations. Currently, many training sets for MLIPs have been built by sampling references from MD trajectories[26,73,144,145,170] which might seem wasteful since only a fraction of a trajectory is useful to the model. Moreover, to achieve good performance on a wide range of thermo-mechanical properties, Bartók *et al.*[118] had to design their training set by carefully introducing hand-crafted configurations. Random structure search has been proposed to automate an efficient exploration of a wide range of configurations in solids[171,172] and active learning techniques to sample relevant configurations from already existing databases of isolated molecules[142] and molecular materials[21] have shown some promise towards building more transferable models. Optimizing the computational cost of obtaining ML predictions is another research direction that has been increasingly important, with several studies highlighting the tradeoffs between cost and accuracy.[65,67,140,141] The extension of ML models beyond potentials, to predict more complex properties, from response tensors to quantum mechanical observables such as the electron charge density or the Hamiltonian is likely to become increasingly important.[15,22–24,123,173–177] As these efforts progress, making ML more useful and accessible, it will become clearer how the incorporation of ML techniques in the atomistic modeling toolbox makes it possible to investigate scientific and technological problems that were inaccessible to electronic structure methods and empirical property models.

Acknowledgements

FM and MC were supported by the NCCR MARVEL, funded by the Swiss National Science Foundation. The authors would also like to thank David Wilkins, Michael Willatt and Andrea Anelli for insightful comments on an early version of this manuscript.

Received: October 22, 2019

[1] X. Song, M. O. Jensen, V. Jogini, R. A. Stein, C. H. Lee, H. S. McHaourab, D. E. Shaw, E. Gouaux, *Nature* **2018**, *556*, 515.
[2] T. A. Pham, Y. Ping, G. Galli, *Nat. Mater.* **2017**, *16*, 401.
[3] L. A. Zepeda-Ruiz, A. Stukowski, T. Oppelstrup, V. V. Bulatov, *Nature* **2017**, *550*, 492.
[4] Y. Shibuta, S. Sakane, E. Miyoshi, S. Okita, T. Takaki, M. Ohno, *Nat. Commun.* **2017**, *8*, 10.
[5] M. A. Rohrdanz, W. Zheng, C. Clementi, *Annu. Rev. Phys. Chem.* **2013**, *64*, 295.
[6] A. Rodriguez, A. Laio, *Science* **2014**, *344*, 1492.
[7] M. Ceriotti, *J. Chem. Phys.* **2019**, *150*, 150901.
[8] S. Curtarolo, G. L. W. Hart, M. B. Nardelli, N. Mingo, S. Sanvito, O. Levy, *Nature Mater*. **2013**, *12*, 191.
[9] L. Ward, C. Wolverton, *Curr. Opin. Solid State Mater. Sci.* **2017**, *21*, 167.
[10] A. C. Mater, M. L. Coote, *J. Chem. Inf. Model.* **2019**, *59*, 2545.
[11] F. A. Faber, O. Anatole von Lilienfeld, in 'Materials Informatics', Wiley, **2019**, pp. 171–179.
[12] P. Gasparotto, R. H. Meißner, M. Ceriotti, *J. Chem. Theor. Comput.* **2018**, *14*, 486.
[13] B. A. Helfrecht, P. Gasparotto, F. Giberti, M. Ceriotti, *Front. Mol. Biosci.* **2019**, *6*, 24.
[14] S. De, A. P. Bartók, G. Csányi, M. Ceriotti, *Phys. Chem. Chem. Phys.* **2016**, *18*, 13754.
[15] F. Musil, S. De, J. Yang, J. E. Campbell, G. M. Day, M. Ceriotti, *Chem. Sci.* **2018**, *9*, 1289.
[16] N. Blöchliger, A. Vitalis, A. Caflisch, *Sci. Rep.* **2014**, *4*, 6264.
[17] L. Kahle, A. Musaelian, N. Marzari, B. Kozinsky, *Phys. Rev. Mater*. **2019**, *3*, 055404.
[18] T. Xie, A. France-Lanord, Y. Wang, Y. Shao-Horn, J. C. Grossman, *Nat. Commun.* **2019**, *10*, 2667.
[19] M. Rupp, A. Tkatchenko, K.-R. Müller, O. A. von Lilienfeld, *Phys. Rev. Lett*. **2012**, *108*, 058301.
[20] A. P. Bartók, M. C. Payne, R. Kondor, G. Csányi, *Phys. Rev. Lett*. **2010**, *104*, 136403.
[21] F. M. Paruzzo, A. Hofstetter, F. Musil, S. De, M. Ceriotti, L. Emsley, *Nat. Commun.* **2018**, *9*, 4501.
[22] F. Brockherde, L. Vogt, L. Li, M. E. Tuckerman, K. Burke, K. R. Müller, *Nat. Commun*. **2017**, *8*, 872.
[23] A. Grisafi, A. Fabrizio, B. Meyer, D. M. Wilkins, C. Corminboeuf, M. Ceriotti, *ACS Cent. Sci.* **2019**, *5*, 57.
[24] C.-I. Wang, M. K. E. Braza, G. C. Claudio, R. B. Nellas, C.-P. Hsu, *J. Phys. Chem. A* **2019**, *123*, 7792.
[25] V. L. Deringer, M. A. Caro, G. Csányi, *Adv. Mater.* **2019**, 1902765.
[26] J. Behler, *J. Chem. Phys.* **2016**, *145*, 170901.
[27] C. Qu, Q. Yu, J. M. Bowman, *Ann. Rev. Phys. Chem.* **2018**, *69*, 151.
[28] Y. Lecun, Y. Bengio, G. Hinton, *Nature* **2015**, *521*, 436.
[29] T. Young, D. Hazarika, S. Poria, E. Cambria, *IEEE Comput. Intell. Mag.* **2018**, *13*, 55.
[30] T. Cohen, M. Geiger, M. Weiler, *ArXiv e-prints* **2018**, arXiv:1811.02017.
[31] C. Esteves, C. Allen-Blanchette, A. Makadia, K. Daniilidis, in 'Computer Vision – ECCV 2018', Eds V. Ferrari, H. C. Sminchisescu, Y. Weiss, Springer International Publishing, Cham, **2018**, pp. 54–70.
[32] N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, P. Riley, *ArXiv e-prints* **2018**, arXiv:1802.08219.
[33] R. Kondor, *ArXiv e-prints* **2018**, arXiv:1803.01588.
[34] M. Weiler, M. Geiger, M. Welling, W. Boomsma, T. Cohen, in 'NIPS'18', Curran Associates Inc., **2018**, pp. 10402–10413, 1807.02547.
[35] R. Kondor, Z. Lin, S. Trivedi, *ArXiv e-prints* **2018**, arXiv:1806.09231.
[36] A. Seko, H. Hayashi, K. Nakayama, A. Takahashi, I. Tanaka, *Phys. Rev. B* **2017**, *95*, 1, arXiv:1611.08645.
[37] A. Mannodi-Kanakkithodi, G. Pilania, T. D. Huan, T. Lookman, R. Ramprasad, *Sci. Rep.* **2016**, *6*, 20952.
[38] D. Xue, P. V. Balachandran, J. Hogden, J. Theiler, D. Xue, T. Lookman, *Nat. Commun.* **2016**, *7*, 11241.
[39] R. M. Martin, 'Electronic Structure', Cambridge University Press, Cambridge, **2004**.
[40] J. Behler, M. Parrinello, *Phys. Rev. Lett*. **2007**, *98*, 146401.
[41] B. J. Braams, J. M. Bowman, *Int. Rev. Phys. Chem.* **2009**, *28*, 577.
[42] Z. Xie, J. M. Bowman, *J. Chem. Theor. Comput.* **2010**, *6*, 26.
[43] A. Shapeev, *Multiscale Model. Sim.* **2016**, *14*, 1153.
[44] F. A. Faber, A. S. Christensen, B. Huang, O. A. Von Lilienfeld, *J. Chem. Phys.* **2018**, *148*, 241717.

[45] K. Hansen, F. Biegler, R. Ramakrishnan, W. Pronobis, O. A. Von Lilienfeld, K. R. Müller, A. Tkatchenko, *J. Phys. Chem. Lett.* **2015**, *6*, 2326.
[46] A. P. Bartók, R. Kondor, G. Csányi, *Phys. Rev. B* **2013**, *87*, 184115.
[47] R. Drautz, *Phys. Rev. B* **2019**, *99*, 014104.
[48] H. Huo, M. Rupp, *ArXiv e-prints* **2017**, arXiv:1704.06439.
[49] A. Samanta, *J. Chem. Phys.* **2018**, *149*, 244102.
[50] A. Seko, A. Togo, I. Tanaka, *Phys. Rev. B* **2019**, *99*, 214108.
[51] M. Hirn, S. Mallat, N. Poilvert, *Multiscale Model. Sim.* **2017**, *15*, 827, arXiv:1605.04654.
[52] M. J. Willatt, F. Musil, M. Ceriotti, *Phys. Chem. Chem. Phys.* **2018**, *20*, 29661.
[53] A. Grisafi, D. M. Wilkins, M. J. Willatt, M. Ceriotti, *ArXiv e-prints* **2019**, arXiv:1904.01623.
[54] A. Grisafi, D. M. Wilkins, G. Csányi, M. Ceriotti, *Phys. Rev. Lett.* **2018**, *120*, 036002.
[55] A. Glielmo, P. Sollich, A. De Vita, *Phys. Rev. B* **2017**, *95*, 214302.
[56] A. Grisafi, M. Ceriotti, *ArXiv e-prints* **2019**, arXiv:1909.04512.
[57] L. Nachbin, 'The Haar integral', R. E. Krieger Pub. Co., **1976**.
[58] M. J. Willatt, F. Musil, M. Ceriotti, *J. Chem. Phys.* **2019**, *150*, 154110.
[59] E. Prodan, W. Kohn, *Proc. Natl. Acad. Sci*. **2005**, *102*, 11635.
[60] G. Galli, M. Parrinello, *Phys. Rev. Lett.* **1992**, *69*, 3547.
[61] S. Goedecker, *Rev. Mod. Phys.* **1999**, *71*, 1085.
[62] M. G. Papadopoulos, R. Zalesny, P. G. Mezey, 'Linear- Scaling Techniques in Computational Chemistry and Physics', Springer Netherlands, Dordrecht, **2011**, p. 536.
[63] D. R. Bowler, T. Miyazaki, *Rep. Prog. Phys.* **2012**, *75*, 1.
[64] M. A. Caro, *Phys. Rev. B* **2019**, *100*, 024112.
[65] T. T. Nguyen, E. Székely, G. Imbalzano, J. Behler, G. Csányi, M. Ceriotti, A. W. Götz, F. Paesani, *J. Chem. Phys.* **2018**, *148*, 241725.
[66] C. Nyshadham, M. Rupp, B. Bekker, A. V. Shapeev, T. Mueller, C. W. Rosenbrock, G. Csányi, D. W. Wingate, G. L. Hart, *npj Comput. Mater.* **2019**, *5*, 51.
[67] Y. Zuo, C. Chen, X. Li, Z. Deng, Y. Chen, J. Behler, G. Csányi, A. V. Shapeev, A. P. Thompson, M. A. Wood, S. P. Ong, *ArXiv e-prints* **2019**, arXiv:1906.08888.
[68] N. Raimbault, A. Grisafi, M. Ceriotti, M. Rossi, *ArXiv e-prints* **2019**, arXiv:1906.07485.
[69] T. Bereau, D. Andrienko, O. A. von Lilienfeld, *J. Chem. Theor. Comput.* **2015**, *11*, 3225.
[70] C. Liang, G. Tocci, D. M. Wilkins, A. Grisafi, S. Roke, M. Ceriotti, *Phys. Rev. B* **2017**, *96*, 041407.
[71] A. Fabrizio, A. Grisafi, B. Meyer, M. Ceriotti, C. Corminboeuf, *Chem. Sci.* **2019**, *10*, 9424.
[72] V. Botu, R. Ramprasad, *Phys. Rev. B* **2015**, *92*, 094306.
[73] Z. Li, J. R. Kermode, A. De Vita, *Phys. Rev. Lett.* **2015**, *114*, 096405.
[74] J. P. Mailoa, M. Kornbluth, S. L. Batzner, G. Samsonidze, S. T. Lam, C. Ablitt, N. Molinari, B. Kozinsky, *ArXiv e-prints* **2019**, arXiv:1905.02791.
[75] M. Ceriotti, M. J. Willatt, G. Csányi, in 'Handbook of Materials Modeling', Springer, Cham, **2018**, pp. 1–27, 1901.10971.
[76] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, K. R. Müller, *Sci. Adv.* **2017**, *3*, e1603015.
[77] K. T. Schütt, H. E. Sauceda, P. J. Kindermans, A. Tkatchenko, K. R. Müller, *J. Chem. Phys.* **2018**, *148*, 241722.
[78] C. M. Bishop, 'Pattern recognition and machine learning', Springer, **2006**, p. 12.
[79] S.-i. Amari, N. Murata, *Neural Comput.* **1993**, *5*, 140.
[80] Tuckerman Mark, 'Statistical Mechanics: Theory and Molecular Simulation', Oxford University Press, **1972**, p. 696.
[81] J. Behler, M. Parrinello, *Phys. Rev. Lett.* **2007**, *98*, 146401.
[82] 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.
[83] F. A. Faber, L. Hutchison, B. Huang, J. Gilmer, S. S. Schoenholz, G. E. Dahl, O. Vinyals, S. Kearnes, P. F. Riley, O. A. von Lilienfeld, *J. Chem. Theory Comput.* **2017**, *13*, 5255.
[84] A. P. Bartók, S. De, C. Poelking, N. Bernstein, J. R. Kermode, G. Csányi, M. Ceriotti, *Sci. Adv.* **2017**, *3*, e1701816.
[85] E. R. Homer, D. M. Hensley, C. W. Rosenbrock, A. H. Nguyen, G. L. W. Hart, *Front. Mater.* **2019**, *6*, 168.
[86] M. Jafary-Zadeh, K. H. Khoo, R. Laskowski, P. S. Branicio, A. V. Shapeev, *J. Alloys Comp.* **2019**, *803*, 1054.
[87] I. I. Novoselov, A. V. Yanilkin, A. V. Shapeev, E. V. Podryabinkin, *Comput. Mater. Sci.* **2019**, *164*, 46.
[88] A. P. Thompson, L. P. Swiler, C. R. Trott, S. M. Foiles, G. J. Tucker, *J. Comput. Phys.* **2015**, *285*, 316.
[89] M. A. Wood, A. P. Thompson, *J. Chem. Phys*. **2018**, *148*, 241721.
[90] Z. Deng, C. Chen, X. G. Li, S. P. Ong, *npj Comput. Mater.* **2019**, *5*, 75.
[91] N. Artrith, T. Morawietz, J. Behler, *Phys. Rev. B* **2011**, *83*, 153101.
[92] N. Artrith, A. Urban, *Comput. Mater. Sci*. **2016**, *114*, 135.
[93] J. S. Smith, O. Isayev, A. E. Roitberg, *Chem. Sci.* **2017**, *8*, 3192.
[94] K. Yao, J. E. Herr, D. W. Toth, R. McKintyre, J. Parkhill, *Chem. Sci*. **2018**, *9*, 2261.
[95] K. Lee, D. Yoo, W. Jeong, S. Han, *Comput. Phys. Commun.* **2019**, *242*, 95.
[96] J. E. Herr, K. Koh, K. Yao, J. Parkhill, *J. Chem. Phys.* **2019**, *151*, 084103.
[97] L. Zhang, J. Han, H. Wang, R. Car, E. Weinan, *Phys. Rev. Lett.* **2018**, *120*, 143001.
[98] A. Khorshidi, A. A. Peterson, *Comput. Phys. Commun.* **2016**, *207*, 310.
[99] N. Artrith, A. Urban, G. Ceder, *Phys. Rev. B* **2017**, *96*, 014112.
[100] K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, A. Tkatchenko, *Nat. Commun.* **2017**, *8*, 6.
[101] N. Lubbers, J. S. Smith, K. Barros, *J. Chem. Phys.* **2018**, *148*, 241715.
[102] O. T. Unke, M. Meuwly, *J. Chem. Theor. Comput.* **2019**, *15*, 3678.
[103] S. K. Natarajan, J. Behler, *Phys. Chem. Chem. Phys.* **2016**, *18*, 28704.
[104] M. Gastegger, J. Behler, P. Marquetand, *Chem. Sci.* **2017**, *8*, 6924.
[105] V. Kapil, J. Behler, M. Ceriotti, *J. Chem. Phys.* **2016**, *145*, 234103.
[106] B. Cheng, E. A. Engel, J. Behler, C. Dellago, M. Ceriotti, *Proc. Nat. Acad. Sci. USA* **2019**, *116*, 1110.
[107] S. D. Huang, C. Shang, P. L. Kang, Z. P. Liu, *Chem. Sci.* **2018**, *9*, 8644.
[108] M. Eckhoff, J. Behler, *J. Chem. Theor. Comput*. **2019**, *15*, 3793.
[109] C. E. Rasmussen, C. K. I. Williams, *Int. J. Neural Sys.* **2006**, *2*, 69, arXiv:026218253X.
[110] R. Ramakrishnan, P. O. Dral, M. Rupp, O. A. Von Lilienfeld, *J. Chem. Theor. Comput.* **2015**, *11*, 2087.
[111] K. Hansen, G. Montavon, F. Biegler, S. Fazli, M. Rupp, M. Scheffler, O. A. Von Lilienfeld, A. Tkatchenko, K. R. Müller, *J. Chem. Theor. Comput*. **2013**, *9*, 3404.
[112] F. A. Faber, A. Lindmaa, O. A. Von Lilienfeld, R. Armiento, *Phys. Rev. Lett.* **2016**, *117*, 135502.
[113] S. De, F. Musil, T. Ingram, C. Baldauf, M. Ceriotti, *J. Cheminformatics* **2017**, *9*, 1.
[114] A. Glielmo, C. Zeni, A. De Vita, *Phys. Rev. B* **2018**, *97*, 184307.
[115] W. J. Szlachta, A. P. Bartók, G. Csányi, *Phys. Rev. B* **2014**, *90*, 104108.
[116] V. L. Deringer, G. Csányi, *Phys. Rev. B* **2017**, *95*, 094203.
[117] V. L. Deringer, N. Bernstein, A. P. Bartók, M. J. Cliffe, R. N. Kerber, L. E. Marbella, C. P. Grey, S. R. Elliott, G. Csányi, *J. Phys. Chem. Lett.* **2018**, *9*, 2879, arXiv:1803.02802.
[118] A. P. Bartók, J. Kermode, N. Bernstein, G. Csányi, *Phys. Rev.* **2018**, *X,* 8.
[119] C. Zeni, K. Rossi, A. Glielmo, Á. Fekete, N. Gaston, F. Baletto, A. De Vita, *J. Chem. Phys.* **2018**, *148*, 241739.
[120] M. Veit, S. K. Jain, S. Bonakala, I. Rudra, D. Hohl, G. Csányi, *J. Chem. Theor. Comput*. **2019**, *15*, 2574, arXiv:1810.10475.
[121] A. Lopez-Bezanilla, O. A. Von Lilienfeld, *Phys. Rev. B* **2014**, *89*, 235411, arXiv:1401.8277v1.
[122] R. Ramakrishnan, M. Hartmann, E. Tapavicza, O. A. Von Lilienfeld, *J. Chem. Phys.* **2015**, *143*, 084111.
[123] A. S. Christensen, F. A. Faber, O. A. von Lilienfeld, *J. Chem. Phys.* **2019**, 150, 064105.
[124] B. Schölkopf, R. Herbrich, A. J. Smola, in 'Computational Learning Theory', Vol. 2111, Eds D. Helmbold, B. Williamson, Springer, Berlin, Heidelberg, **2001**, pp. 416–426.
[125] A. P. Bartók, G. Csányi, *Int. J. Quantum Chem*. **2015**, *115*, 1051.
[126] J. Quiñonero-Candela, C. E. Rasmussen, *J. Machine Learning Res.* **2005**, *6*, 1939.
[127] H. Liu, Y.-S. Ong, X. Shen, J. Cai, *ArXiv e-prints* **2018**, arXiv:1807.01065.
[128] M. Seeger, C. K. I. Williams, N. D. Lawrence, in 'Workshop on AI and Statistics', **2003**, p. 9.
[129] F. Musil, M. J. Willatt, M. A. Langovoy, M. Ceriotti, *J. Chem. Theor. Comput.* **2019**, *15*, 906.
[130] A. J. Smola, P. Bartlett, *Adv. Neural Inf. Process. Sys.* **2001**, *13*, 619.
[131] S. Sathiya Keerthi, W. Chu, *Adv. Neural Inf. Processing Sys.* **2005**, 643.
[132] J. Schreiter, D. Nguyen-Tuong, M. Toussaint, *Neurocomputing* **2016**, *192*, 29.
[133] M. Ceriotti, G. A. Tribello, M. Parrinello, *J. Chem. Theory Comput.* **2013**, *9*, 1521.
[134] G. Imbalzano, A. Anelli, D. Giofré, S. Klees, J. Behler, M. Ceriotti, *J. Chem. Phys.* **2018**, *148*, 241730.
[135] O. A. von Lilienfeld, *Angew. Chem. Int. Ed*. **2018**, *57*, 4164.
[136] C. J. Pickard, F. Mauri, *Phys. Rev. B* **2001**, *63*, 2451011.
[137] J. R. Yates, C. J. Pickard, F. Mauri, *Phys. Rev. B* **2007**, *76*, 024401.
[138] E. Salager, G. M. Day, R. S. Stein, C. J. Pickard, B. Elena, L. Emsley, *J. Am. Chem. Soc*. **2010**, *132*, 2564.
[139] F. M. Paruzzo, A. Hofstetter, F. Musil, S. De, M. Ceriotti, L. Emsley, 'ShiftML Website', *http://shiftml.org*, **2018**.
[140] W. Li,Y. Ando, *Phys. Chem. Chem. Phys.* **2018**, *20*, 30006.
[141] A. Kamath, R. A. Vargas-Hernández, R. V. Krems, T. Carrington, S. Manzhos, *J. Chem. Phys.* **2018**, *148*, 241702.
[142] J. S. Smith, B. Nebgen, N. Lubbers, O. Isayev, A. E. Roitberg, *J. Chem. Phys.* **2018**, *148*, 241733.
[143] K. Gubaev, E. V. Podryabinkin, A. V. Shapeev, *J. Chem. Phys.* **2018**, *148*, 241727.

[144] J. Vandermause, S. B. Torrisi, S. Batzner, A. M. Kolpak, B. Kozinsky, *ArXiv e-prints* **2019**, arXiv:1904.02042.
[145] R. Jinnouchi, F. Karsai, G. Kresse, *Phys. Rev. B* **2019**, *100*, 014105.
[146] D. J. Mackay, 'Probable networks and plausible predictions - a review of practical bayesian methods for supervised neural networks', **1995**, *http://www.inference.org.uk/mackay/network.pdf*.
[147] C. Blundell, J. Cornebise, K. Kavukcuoglu, D. Wierstra, *ArXiv e-prints* **2015**, arXiv:1505.05424.
[148] T. G. Dietterich, 'Ensemble Methors in Machine Learning', in MCS 2000: Multiple Classifier Systems', Springer, Berlin, Heidelberg, **2000**, pp. 1–15.
[149] A. A. Peterson, R. Christensen, A. Khorshidi, *Phys. Chem. Chem. Phys.* **2017**, *19*, 10978.
[150] J. Behler, *Angew. Chem. Int. Ed.* **2017**, *56*, 12828.
[151] B. Efron, *Annals of Statistics* **1979**, *7*, 1.
[152] D. Politis, J. P. Romano, M. Wolf, *J. Statistical Planning and Inference* **1999**, *79*, 179.
[153] D. N. Politis, J. P. Romano, *Annals of Statistics* **1994**, *22*, 2031.
[154] R. Tibshirani, *Neural Comput.* **1996**, *8*, 152.
[155] M. Segù, A. Loquercio, D. Scaramuzza, *ArXiv e-prints* **2019**, arXiv:1907.06890.
[156] J. P. Janet, C. Duan, T. Yang, A. Nandy, H. J. Kulik, *Chem. Sci.* **2019**, *10*, 7913.
[157] S. Ryu, Y. Kwon, W. Y. Kim, *Chem. Sci.* **2019**, *10*, 8438.
[158] A. Seko, A. Takahashi, I. Tanaka, *Phys. Rev. B* **2015**, *92*, 054113, arXiv:1505.03994.
[159] T. Xie and J. C. Grossman, *J. Chem. Phys*. **2018**, *149*, 174111.
[160] J. Schmidt, J. Shi, P. Borlido, L. Chen, S. Botti, M. A. Marques, *Chem. Mater.* **2017**, *29*, 5090.
[161] H. Wang, L. Zhang, J. Han, E. Weinan, *Comp. Phys. Commun.* **2018**, *228*, 178.
[162] A. S. Abbott, J. M. Turney, B. Zhang, D. G. A. Smith, D. Al-tarawy, H. F. Schaefer, *J. Chem. Theor. Comput.* **2019**, *15*, 4386.
[163] K. T. Schütt, P. Kessel, M. Gastegger, K. A. Nicoli, A. Tkatchenko, K. R. Müller, *J. Chem. Theor. Comput.* **2019**, *15*, 448,, arXiv:1809.01072.
[164] A. Singraber, J. Behler, C. Dellago, *J. Chem. Theor. Comput.* **2019**, *15*, 1827.
[165] M. Gastegger, P. Marquetand, *ArXiv e-prints* **2018**, arXiv:1812.07676.
[166] M. S. Jørgensen, M. N. Groves, B. Hammer, *J. Chem. Theor. Comput*. **2017**, *13*, 1486.
[167] E. V. Podryabinkin, E. V. Tikhonov, A. V. Shapeev, A. R. Oganov, *Phys. Rev. B* **2019**, *99*, 064114.
[168] T. Morawietz, A. Singraber, C. Dellago, J. Behler, *Proc. Natl. Acad. Sci. U. S. A.* **2016**, *113*, 8368.
[169] B. Cheng, E. A. Engel, J. Behler, C. Dellago, M. Ceriotti, *Proc. Natl. Acad. Sci. U. S. A.* **2019**, *116*, 1110.
[170] T. D. Huan, R. Batra, J. Chapman, S. Krishnan, L. Chen, R. Ramprasad, *npj Comput. Mater.* **2017**, *3*, 37.
[171] V. L. Deringer, C. J. Pickard, G. Csányi, *Phys. Rev. Lett.* **2018**, *120*, 156001, arXiv:1710.10475.
[172] N. Bernstein, G. Csányi, V. L. Deringer, *ArXiv e-prints* **2019**, arXiv:1905.10407.
[173] K. T. Schütt, M. Gastegger, A. Tkatchenko, K. R. Müller, R. J. Maurer, *ArXiv e-prints* **2019**, arXiv:1906.10033.
[174] E. Schmidt, A. T. Fowler, J. A. Elliott, P. D. Bristowe, *Comput. Mater. Sci.* **2018**, *149*, 250.
[175] T. Zubatyuk, B. Nebgen, N. Lubbers, J. S. Smith, R. Zubatyuk, G. Zhou, C. Koh, K. Barros, O. Isayev, S. Tretiak, *ArXiv e-prints* **2019**, arXiv:1909.12963.
[176] D. M. Wilkins, A. Grisafi, Y. Yang, K. U. Lao, R. A. DiStasio, M. Ceriotti, *Proc. Natl. Acad. Sci. U. S. A.* **2019**, *116*, 3401.
[177] H. Ji, Y. Jung, *J. Chem. Phys.* **2018**, *148*, 241742.