# q-pac: A Python Package for Machine Learned Charge Equilibration Models

Martin Vondrák,[1] Karsten Reuter,[1] and Johannes T. Margraf[1, a)]

*Fritz-Haber-Institut der Max-Planck-Gesellschaft, Faradayweg 4-6, D-14195 Berlin, Germany*

Many state-of-the art machine learning (ML) interatomic potentials are based on a local or semi-local (message-passing) representation of chemical environments. They therefore lack a description of long-range electrostatic interactions and non-local charge transfer. In this context, there has been much interest in developing ML-based charge equilibration models, which allow the rigorous calculation of long-range electrostatic interactions and the energetic response of molecules and materials to external fields. The recently reported kQEq method achieves this by predicting local atomic electronegativities using Kernel ML. This paper describes the q-pac Python package, which implements several algorithmic and methodological advances to kQEq and provides an extendable framework for the development of ML charge equilibration models.

[a)] Electronic mail: margraf@fhi.mpg.de

## I. INTRODUCTION

Atomistic machine learning (ML) methods and interatomic potentials in particular have had an enormous impact in the fields of molecular and materials simulation.[1–5] One of the key innovations that made this possible was the idea of decomposing the total energy of a system into atomic contributions, which could be learned as a function of each atom's chemical environment within a certain cutoff radius using Neural Networks (NNs)[6] or Gaussian Process Regression (GPR)[7]. This locality assumption has enabled the construction of highly accurate, computationally efficient and size-extensive potentials, that approach first-principles accuracy at a fraction of the cost.[8–11]

At the same time, locality ultimately also limits the achievable accuracy of a potential, since information beyond the cutoff radius is not taken into account in this case.[12–14] Indeed, long-range interactions can be substantial in bulk systems, most prominently due to the Coulomb interaction, which decays slowly ($\sim \frac{1}{r}$) with interatomic distance. These electrostatic interactions are often screened in practice, so that local potentials can still effectively describe polar solids and liquids with surprising accuracy.[9,10,14] Unfortunately, this cannot always be relied upon, however. For example, non-local charge transfer at heterogeneous interfaces or through molecular wires cannot be adequately captured in this manner.[15] Similarly, the relative stability of molecular crystal polymorphs sensitively depends on a balance of long-ranged electrostatic and dispersion interactions, precluding a purely local description.[16,17]

Due to these limitations, the inclusion of long-range interactions in ML potentials has been an active field of study, with several different approaches in use. These for example include the use of global or non-local descriptors.[18–20] In many cases, physical baseline models can also provide the correct long-range physics at affordable computational cost ($\Delta$-ML).[16,17,21–23] Finally, message-passing neural networks can extend the range of local interatomic potentials by a multiple of the employed cutoff, though without including the full long-range interactions present in a periodic system.[24]

In this contribution, we focus on approaches that tackle the problem of long-range electrostatics by describing the charge distribution of molecules or materials within the ML model itself (*e.g. via* partial charges). This has the advantage that it allows incorporating different total charge states and the response to external fields rigorously. The simplest approach to this end is to directly learn suitable reference charges, *e.g.* from Hirshfeld decomposition.[25,26] While this in principle affords a reasonable description of long-range electrostatics, it does not resolve the issue of non-local charge transfer, since the charges are in this case themselves functions of a local ML model.

To overcome this, Goedecker and co-workers proposed the Charge Equilibration via Neural Network Technique (CENT),[27–29] where the charges are obtained by minimizing a charge dependent energy function. Specifically, CENT uses the classical Charge Equilibration (QEq) model[30] as a basis, replacing fixed elemental electronegativities by environment-dependent ones, predicted by a NN. This approach was subsequently developed further by Behler, Goedecker and co-workers into the Fourth Generation High Dimensional Neural Network Potentials (4GHDNNP).[1,15,31] Here, CENT and local NN potentials are combined and partial charges are fitted to reproduce those obtained from Hirshfeld partitioning.[32] Similarly, Xie, Person and Smalls reported a self-consistent NN potential, where charges are obtained through the gradient-based minimization of a coupled local and electrostatic energy function.[33] Here, charges from Becke population analysis were used as a reference for the partial charges.

Our recently reported Kernel Charge Equilibration (kQEq) method is in the same spirit as these approaches but uses Kernel ML instead of NNs.[34] Kernel methods are frequently used for interatomic potentials, as they are highly data-efficient, depend on few hyperparameters, and can be trained through a closed-form linear algebra expression.[4] Furthermore, kQEq avoids the ambiguity of charge partitioning schemes by training directly on electrostatic observables, such as the dipole moment.

In this paper, we introduce the `q-pac` Python package. `q-pac` provides a modular framework for implementing machine-learned charge equilibration methods, with a particular focus on kQEq. We review the kQEq methodology and describe several new algorithmic and methodological advances in `q-pac`. In particular, the Kernel Ridge Regression (KRR) approach of the original kQEq paper is replaced by a sparse GPR formulation, which provides better computational scaling of training and prediction as a function of the training set size. Furthermore, additional fitting targets and the possibility to fit multiple properties at the same time have been implemented. Notably, this includes energies, which allows the development of fully long-ranged ML interatomic potentials based on kQEq. Finally, some example applications of these new capabilities are showcased.

## II. THEORY

To provide a consistent account of the methodology, the kQEq working equations are rederived in this section, starting from the classical QEq approach of Rappe and Goddard[30,35]. Differences and new features relative to the original implementation presented in Ref. 34 are highlighted where appropriate. Atomic units are used in all equations.

**Charge Equilibration:** The core idea of QEq[30,35] and related methods is to define a simple energy expression that depends on the charge distribution within a system. The ground-state charge distribution for a given geometry is then obtained by minimizing this energy, under the constraint that the total charge is conserved.

The charge distribution in a molecule or solid is rigorously described by the electron density $\rho(\mathbf{r})$ and the location of the nuclei. Since the electron density is a complex three dimensional distribution, it is computationally convenient to work with a more simplified representation such as atomic partial charges, however. To this end, we can split the total electron density into a reference density $\rho_0(\mathbf{r})$ and a fluctuation term $\delta\rho(\mathbf{r})$, where the former is typically the superposition of electron densities of the corresponding isolated spherical atoms (see Fig. 1). Together with the corresponding nuclei, these atomic reference densities are charge neutral and therefore do not contribute to the long-range electrostatic interactions, leaving the fluctuation density as the object of interest.

In the following, we will assume some atomic partitioning of the fluctuation density

$$\delta\rho(\mathbf{r}) = \sum_i \delta\rho_i(\mathbf{r}), \tag{1}$$

where $\delta\rho_i(\mathbf{r})$ is the local fluctuation density around atom $i$. This allows us to define partial charges as:

$$q_i = \int \delta\rho_i(\mathbf{r})d\mathbf{r}, \tag{2}$$

Note that since there is no unique partitioning of $\delta\rho(\mathbf{r})$, the partial charges are also to some extent arbitrary, although canonical choices like Hirshfeld partitioning exist.

We can now approximate the total energy of a non-periodic system as

$$E_{\text{tot}} \approx E_0 + E_{\text{QEq}} = E_0 + \underbrace{\sum_{i=1}^{N}\left(\chi_i q_i + \frac{1}{2}J_i q_i^2\right)}_{\text{Site}-\text{Energy}} + \underbrace{\frac{1}{2}\iint \frac{\delta\rho(\mathbf{r})\delta\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}d\mathbf{r}d\mathbf{r}'}_{\text{Coulomb}-\text{Integral}}. \tag{3}$$

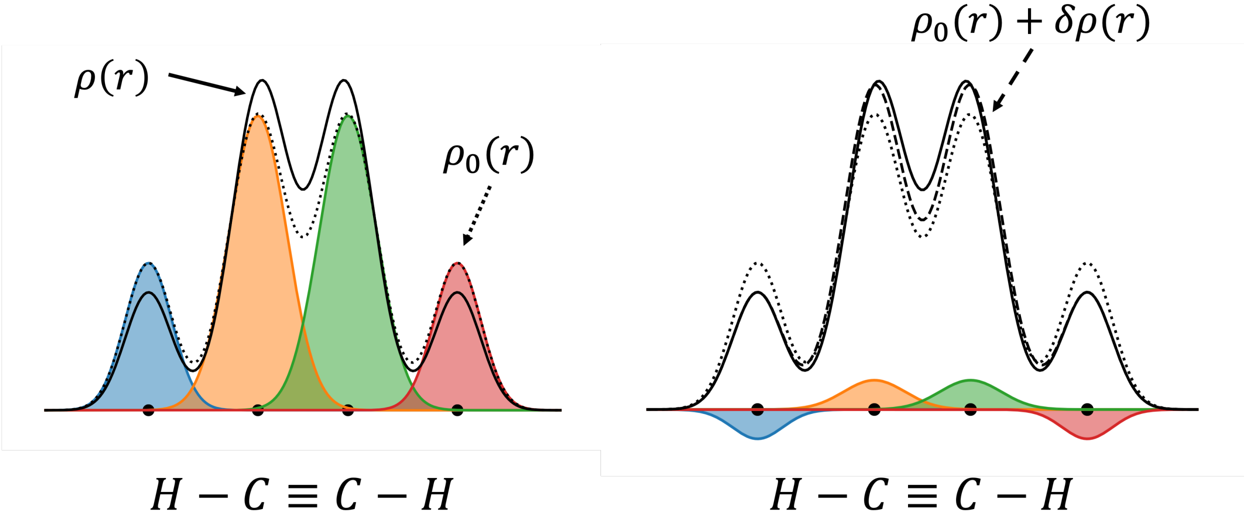

FIG. 1. Illustration of the approximate electron density decomposition into an isolated atom density $\rho_0(\mathbf{r})$ (left) and fluctuation density $\delta\rho(\mathbf{r})$ (right), for a schematic one-dimensional acetylene molecule. The solid line represents the target electron density $\rho(r)$, the dotted line the superposition of isolated atom densities $\rho_0(r)$, and the dashed line a combination of $\rho_0(r)$ with an approximate fluctuation density $\delta\rho(r)$. The latter describes charge transfer and polarization within the molecule.

Here, the first term $E_0$ is a charge-independent reference energy. This could, *e.g.*, be defined as the sum of the energies of the isolated neutral atoms or stem from a local (charge-independent) interatomic potential. The second term $E_{\text{QEq}}$ collects all charge-dependent terms and will be our primary focus in the following. $E_{\text{QEq}}$ can itself further be divided into two terms. The first of these is a site-energy term that sums over all $N$ atoms $i$ and represents the second-order Taylor expansion of the atomic energy with respect to the partial charges. In this context, the expansion coefficient $\chi_i$ is usually termed the electronegativity, while $J_i$ is the electronic hardness. In the original QEq scheme, both of these coefficients are element-dependent parameters. The second term in $E_{\text{QEq}}$ is the classical Coulomb energy of the fluctuation density.

In order to evaluate $E_{\text{QEq}}$ we now need to define a mathematical expression of the fluctuation density and its partitioning. Specifically, we will assume that the fluctuation density $\delta\rho(\mathbf{r})$ can approximately be expressed as a superposition of spherically symmetric atom-centered Gaussians. Each of these Gaussians is normalized to the corresponding atomic partial charge $q_i$ according to Eq.2 and has an inverse distribution width $\phi_i = 1/(2\alpha_i)^{1/2}$, where $\alpha_i$ can be interpreted as an atomic radius. This leads to the expression

$$\delta\rho(\mathbf{r}) \approx \sum_{i=1}^{N} -q_i \left(\frac{\phi_i}{\sqrt{\pi}}\right)^3 \exp\left(-\phi^2 |\mathbf{r} - \mathbf{r}_i|^2\right). \tag{4}$$

Here, the Gaussians are centered at the atomic positions $\mathbf{r}_i$. Using this definition, the Coulomb integral in Eq. 3 can be evaluated analytically as:

$$\iint \frac{\delta\rho(\mathbf{r})\delta\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r} d\mathbf{r}' = \sum_{i=1}^{N} \left( q_i^2 \frac{1}{2\alpha_i\sqrt{\pi}} + \sum_{j=1}^{N} q_i q_j \frac{\operatorname{erf}\left(\frac{r_{ij}}{\sqrt{2}\gamma_{ij}}\right)}{r_{ij}} \right), \tag{5}$$

with $\gamma_{ij} = \sqrt{(\alpha_i^2 + \alpha_j^2)}$ and $r_{ij}$ being the distance between atoms $i$ and $j$. With this, $E_{\mathrm{QEq}}$ from Eq. 3 can be rewritten as

$$E_{\mathrm{QEq}} = \sum_{i=1}^{N} \left[ \chi_i q_i + \frac{1}{2}\left( J_i + \frac{1}{\alpha_i\sqrt{\pi}} \right) q_i^2 \right] + \frac{1}{2} \sum_{i,j}^{N} q_i q_j \frac{\operatorname{erf}\left(\frac{r_{ij}}{\sqrt{2}\gamma_{ij}}\right)}{r_{ij}}. \tag{6}$$

Note that here the on-site contribution to the Coulomb integral has been pulled into the electronic hardness term. We can thus interpret the parameter $J_i$ as a non-classical contribution to the hardness, while the classical contribution is given by the self-energy of the Gaussians.

In order to obtain the equilibrium partial charges $q_i$, $E_{\mathrm{QEq}}$ must now be minimized. Due to the chosen functional form of the site energy, this expression is quadratic in $q_i$, so that the optimal charges can be computed in closed form. To this end, we take the derivative of Eq. 6 with respect to each partial charge and set it to zero. This leads to the linear system of equations:

$$\frac{\partial E_{\mathrm{QEq}}}{\partial q_i} = \sum_{j=1}^{N} A_{ij} q_j + \chi_i = 0 \tag{7}$$

with $A_{ij}$ being the elements of the hardness matrix $\mathbf{A}$ defined as:

$$A_{ij} = \begin{cases} J_i + \frac{1}{\alpha_i\sqrt{\pi}} & \text{for } i = j, \\ \frac{\operatorname{erf}\left(\frac{r_{ij}}{\sqrt{2}\gamma_{ij}}\right)}{r_{ij}} & \text{otherwise.} \end{cases} \tag{8}$$

Using a Lagrange multiplier $\lambda$ to conserve the total charge $q_{\mathrm{tot}}$, we obtain a linear system of equations that can be expressed in matrix notation as

$$\underbrace{\begin{pmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,N} & 1 \\ A_{2,1} & A_{2,2} & \cdots & A_{2,N} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A_{N,1} & A_{N,2} & \cdots & A_{N,N} & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{pmatrix}}_{\bar{\mathbf{A}}} \cdot \underbrace{\begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_N \\ \lambda \end{pmatrix}}_{\bar{\mathbf{q}}} = - \underbrace{\begin{pmatrix} \chi_1 \\ \chi_2 \\ \vdots \\ \chi_N \\ -q_{\text{tot}} \end{pmatrix}}_{\bar{\boldsymbol{\chi}}} \tag{9}$$

Here the bars denote that these arrays are expanded by one dimension due to the Lagrange multiplier. Without the bars, these symbols represent the corresponding $N$-dimensional arrays.

The charge vector $\bar{\mathbf{q}}$ (including the Lagrange multiplier $\lambda$) can now easily be computed as

$$\bar{\mathbf{q}} = -\bar{\mathbf{A}}^{-1}\bar{\boldsymbol{\chi}}, \tag{10}$$

meaning that the charges are a linear function of the electronegativities. Using this matrix notation, $E_{\text{QEq}}$ can be expressed as

$$E_{\text{QEq}} = \frac{1}{2}\mathbf{q}^T\mathbf{A}\mathbf{q} + \mathbf{q}^T\boldsymbol{\chi}. \tag{11}$$

**Periodic Boundary Conditions:** Up to this point (and in Ref. 34) we have only considered systems in open boundary conditions (*i.e.* isolated molecules in the gas phase). For periodic systems, Eqs. 10 and 11 can also be used. However, this requires using Ewald summation in the construction of the hardness matrix $\mathbf{A}$, in order to take the full long-range interactions in an infinite crystal into account.[36] In particular, the Coulomb integral must be modified. The same implementation as in 31 was adopted here.

Ewald summation allows the efficient computation of the electrostatic energy of $N$ point charges in periodic boundary conditions by separating it into a real-space and a reciprocal-space contribution. To this end, each charge is embedded into an auxiliary Gaussian charge distribution of the opposite sign and width $\eta$, defined as:[37]

$$\eta = \frac{1}{\sqrt{2}\pi}V^{\frac{1}{3}}, \tag{12}$$

where $V$ is a volume of the unit cell. Note that these auxiliary Gaussians are not to be confused with the ones defined in Eq. 4.

In the long-range, the electrostatic interactions between the point charges and the auxiliary charge distributions cancel out, so that the short-range part of the electrostatic energy $E_{\text{real}}$ can be evaluated in real space as

$$E_{\text{real}} = \frac{1}{2}\sum_{i=1}^{N}\sum_{j\neq i}^{N_{\text{neig}}} q_i q_j \frac{\operatorname{erfc}(\frac{r_{ij}}{\sqrt{2}\eta})}{r_{ij}}. \tag{13}$$

Here, the first sum goes over the $N$ atoms $i$ in the unit cell and the second sum goes over all $N_{\text{neig}}$ atoms $j$ (including periodic replicas) within the cutoff distance $r_{\text{real}}$ of atom $i$. The cutoff is derived from the width $\eta$ of the auxiliary Gaussians, and depends on the desired accuracy $\epsilon$, which is a small positive number determined by the user (as in Ref. 31, we use a default value of $\epsilon = 10^{-8}$). This yields the following expression for $r_{\text{real}}$:[37]

$$r_{\text{real}} = \sqrt{2}\eta\sqrt{\log\epsilon} \tag{14}$$

As a second step, the long-range interactions of the auxiliary charge distributions is computed. This can be evaluated efficiently in reciprocal space using the Fourier transform of the auxiliary charge density:

$$E_{\text{recip}} = \frac{2\pi}{V}\sum_{\mathbf{k}\neq 0}\frac{\exp\left(\frac{-\eta^2|\mathbf{k}|^2}{2}\right)}{|\mathbf{k}|^2}\left(\sum_{i=1}^{N} q_i \exp(i\mathbf{k}\cdot\mathbf{r}_i)\right)^2 \tag{15}$$

Here, the first sum goes over all reciprocal lattice points $\mathbf{k}$ within the cutoff $r_{\text{recip}}$, which is computed as

$$r_{\text{recip}} = \frac{\sqrt{2}}{\eta}\sqrt{\log\epsilon}. \tag{16}$$

The cutoff distance $r_{\text{recip}}$ depends again on the user defined accuracy parameter $\epsilon$.

Finally, the the self-interaction of the auxiliary Gaussians charges is accounted for via

$$E_{\text{self}} = -\sum_{i=1}^{N}\frac{q_i^2}{\sqrt{2\pi}\eta}. \tag{17}$$

Summation of all previous terms is equal to the electrostatic energy of $N$ point charges in periodic boundary conditions:

$$E_{\text{Ewald}} = E_{\text{real}} + E_{\text{recip}} + E_{\text{self}}. \tag{18}$$

Because we use Gaussian charge distributions of width $\alpha_i$ instead of point charges an additional correction term is required[36]:

$$E_{\text{Gauss}} = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j\neq i}^{N_{\text{neig}}} q_i q_j \frac{\operatorname{erfc}\left(\frac{r_{ij}}{\sqrt{2}\gamma_{ij}}\right)}{r_{ij}} + \sum_{i=1}^{N}\frac{q_i^2}{2\sqrt{\pi}\alpha_i}, \tag{19}$$

Here, the first term is again applied for all interactions within the cutoff $r_{\text{real}}$, while the second term corresponds to the on-site contribution of the Coulomb integral which is also present in the non-periodic case.

The periodic Coulomb integral can now be written as:

$$\iint \frac{\delta\rho(\mathbf{r})\delta\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}d\mathbf{r}' = E_{\text{Ewald}} - \frac{1}{2}\sum_{i=1}^{N}\sum_{j\neq i}^{N_{\text{neig}}} q_i q_j \frac{\operatorname{erfc}\left(\frac{r_{ij}}{\sqrt{2}\gamma_{ij}}\right)}{r_{ij}} + \sum_{i=1}^{N}\frac{q_i^2}{2\sqrt{\pi}\alpha_i} \tag{20}$$

From this, the periodic hardness matrix elements can be derived analogously to the non-periodic case.

**Kernel Charge Equilibration:** As described so far, conventional charge equilibration schemes like QEq require the definition of three parameters per element. These are the electronegativity ($\chi_i$), the non-classical contribution to the hardness ($J_i$) and the atomic radius ($\alpha_i$). In practice, this limits the achievable accuracy of QEq, since the same electronic properties are assumed for all atoms of the same element, independent of their chemical environment and oxidation state. ML based charge equilibration methods can overcome this limitation by allowing $\chi_i$ (and in principle also the other parameters) to adapt to the environment of each atom, *e.g.* via a NN.[27,28]

To implement this environment dependence in a Kernel ML framework, kQEq expresses the electronegativities in terms of atomic environment representation vectors $\mathbf{p}_i$, a kernel function $k$ and regression weights $w_m$ as:

$$\chi_i(\mathbf{p}_i) = \sum_{m=1}^{M} k(\mathbf{p}_i, \mathbf{p}_m) w_m, \tag{21}$$

where the sum goes over all atoms $m$ in a representative set of chemical environments. Simply put, this equation thus assigns the electronegativity of atom $i$ based on the similarity between $i$ and each atom $m$ in the representative set, as quantified by the kernel function $k$.

In the original kQEq implementation reported in Ref. 34 the representative set simply consisted of all chemical environments in the training set. In this case, the cost of predicting the electronegativities scales linearly with the number of training samples. Even worse, the training cost of such a Kernel Ridge Regression (KRR) model scales cubically with the number of training samples. The new implementation in `q-pac` therefore uses a different regression framework, namely sparse GPR. This is directly analogous to the approach used in Gaussian Approximation Potentials (GAP).[4,7] Specifically, the representative set now consists of $M$ environments (also called sparse points), which form a representative subset of the training set. As in GAP, these are selected through a CUR decomposition of the matrix of representation vectors.[4,38]

To represent the chemical environments of atoms, Smooth Overlap of Atomic Positions (SOAP)[39] vectors are used, as implemented in the `Dscribe` package. As in GAP, polynomial kernels are used to quantify similarities between SOAP vectors

$$k(\mathbf{p}_i, \mathbf{p}_j) = (\mathbf{p}_i \cdot \mathbf{p}_j)^{\zeta}. \tag{22}$$

Throughout this manuscript, $\zeta = 2$ is used as a default. SOAP vectors are normalized so that $k(\mathbf{p}_i, \mathbf{p}_i) = 1$.

To predict the electronegativities of $N$ atoms, Eq. 21 can be rewritten as a matrix-vector multiplication:

$$\boldsymbol{\chi} = \mathbf{K}_{NM}\mathbf{w} \tag{23}$$

Here and in the following, we use the notation of Csányi and co-workers for Kernel matrices,[4] where the subscripts indicate their dimensions. $\mathbf{K}_{NM}$ is thus a matrix containing the evaluations of the kernel function between all $M$ sparse points and all $N$ environments to be predicted. The thus obtained electronegativities can then be used to predict charges via Eq. 10.

**Training on Electrostatic Properties:** In principle, Eq. 23 and Eq. 10 fully specify the kQEq method. However, this leaves the key question of what the regression weights $w_m$ should be. In Ref. 34, we showed that these can be computed in closed-form by solving a regularized least-squares problem. This was demonstrated by fitting kQEq models on molecular dipole moments. Importantly, the fact that training can be performed as a single closed-form linear algebra operation in this case hinges on the fact that dipole moments are linear functions of atomic partial charges. Additionally, the charges in QEq are linear functions of the electronegativities (see Eq. 10) and the electronegativities are linear functions of the regression weights (see Eq. 23). Indeed, Kernel methods like KRR and GPR allow arbitrary linear transformations of the regression output in the loss function.

Taking advantage of this, `q-pac` provides a generalized loss function for fitting to any electrostatic property that is a linear function of atomic partial charges. The corresponding regularized least-squares loss reads:

$$\mathcal{L}_t = ||\mathbf{T}_t\bar{\mathbf{q}} - \mathbf{t}_{\text{ref}}||^2_{\boldsymbol{\Sigma}_t^{-1}} + ||\mathbf{w}||^2_{\mathbf{K}_{MM}} = (\mathbf{T}_t\bar{\mathbf{q}} - \mathbf{t}_{\text{ref}})^T\boldsymbol{\Sigma}_t^{-1}(\mathbf{T}_t\bar{\mathbf{q}} - \mathbf{t}_{\text{ref}}) + \mathbf{w}^T\mathbf{K}_{MM}\mathbf{w}, \tag{24}$$

where $\mathbf{t}_{\text{ref}}$ is a general target property (*e.g.* atomic charges or dipole vector elements) and $\mathbf{T}_t$ is a transformation matrix that converts a Lagrange multiplier expanded vector (see Eq. 25) of charges $\bar{\mathbf{q}}$ to the target property. $\mathbf{\Sigma}_t$ is an $N$-dimensional diagonal regularization matrix that contains noise parameters $\sigma_i^2$, which are proportional to the regularization strength. Unlike the unit-less regularization parameter in the previous KRR implementation, $\sigma_i$ has the unit of the predicted property and can be interpreted as the expected accuracy of the fit. Furthermore, the GPR framework allows assigning individual values of $\sigma_i$ in order to weight training samples differently. For simplicity, a single value of $\sigma_i$ is used for each of the examples below.

We now aim to find the vector of weights $\mathbf{w}$ which minimizes this loss function. To this end, Eq. 24 must be rewritten so that it only depends on $\mathbf{w}$. Here, a technical difficulty arises, in that Eq. 23 yields a vector of electronegativities $\boldsymbol{\chi}$, while Eq. 10 requires the extended vector $\bar{\boldsymbol{\chi}}$, which includes the total charge of the system $q_{\text{tot}}$. This transformation is achieved via an auxiliary matrix $\mathbf{X}$:

$$\underbrace{\begin{pmatrix} \chi_1 \\ \chi_2 \\ \vdots \\ \chi_N \\ -q_{\text{tot}} \end{pmatrix}}_{\bar{\boldsymbol{\chi}}} = \underbrace{\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \end{pmatrix}}_{\mathbf{X}} \cdot \underbrace{\begin{pmatrix} K_{1,1} & K_{1,2} & \cdots & K_{1,M} \\ K_{2,1} & K_{2,2} & \cdots & K_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ K_{N,1} & K_{N,2} & \cdots & K_{N,M} \end{pmatrix}}_{\mathbf{K}_{NM}} \cdot \underbrace{\begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_M \end{pmatrix}}_{\mathbf{w}} - \underbrace{\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ q_{\text{tot}} \end{pmatrix}}_{\mathbf{q}_{\text{tot}}} \tag{25}$$

Plugging this into Eq. 10 yields

$$\bar{\mathbf{q}} = -\bar{\mathbf{A}}^{-1}\bar{\boldsymbol{\chi}} = -\bar{\mathbf{A}}^{-1}(\overbrace{\mathbf{X}\,\underbrace{\mathbf{K}_{NM}\mathbf{w}}_{\boldsymbol{\chi}} - \mathbf{q}_{\text{tot}}}^{\bar{\boldsymbol{\chi}}}) \tag{26}$$

Finally, the transformation matrix $\mathbf{T}_t$ determines the targeted property $\mathbf{t}$.

$$\mathbf{t} = \mathbf{T}_t\bar{\mathbf{q}} = -\mathbf{T}_t\bar{\mathbf{A}}^{-1}(\mathbf{X}\mathbf{K}_{NM}\mathbf{w} - \mathbf{q}_{\text{tot}}) \tag{27}$$

In the current implementation, transformation matrices for charges and dipoles are provided:

$$\mathbf{T}_q = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & 0 \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \tag{28}$$

$$\mathbf{T}_\mu = \begin{pmatrix} x_1 & x_2 & \cdots & x_N & 0 \\ y_1 & y_2 & \cdots & y_N & 0 \\ z_1 & z_2 & \cdots & z_N & 0 \end{pmatrix} \tag{29}$$

Note that these transformation matrices can in principle easily be modified to accommodate for higher multipole moments or other charge-derived electrostatic properties such as electrostatic potentials at given grid points.

The final form of the loss function is obtained by plugging Eq. 27 into Eq. 24, yielding:

$$\mathcal{L}_t = ||\mathbf{t} - \mathbf{t}_{\text{ref}}||^2_{\mathbf{\Sigma}_t^{-1}} + ||\mathbf{w}||^2_{\mathbf{K}_{MM}} = ||\mathbf{T}_t\mathbf{A}^{-1}(\mathbf{X}\mathbf{K}_{NM}\mathbf{w} - \mathbf{q}_{\text{tot}}) - \mathbf{t}_{\text{ref}}||^2_{\mathbf{\Sigma}_t^{-1}} + ||\mathbf{w}||^2_{\mathbf{K}_{MM}} \tag{30}$$

In this loss function, both the least-squares and regularization terms are quadratic in the regression weights $\mathbf{w}$. The optimal weights are obtained by taking the derivative $\nabla_{\mathbf{w}}\mathcal{L}$ (which is linear in $\mathbf{w}$), setting it to zero and solving for $\mathbf{w}$.

While Eq. 30 is a general loss function for single-property prediction, it may also be of interested to train models on multiple properties simultaneously. To this end, `q-pac` also allows combined loss functions. For instance, one may want to fit a model that reproduces molecular dipoles with partial charges that are close to some population analysis scheme:

$$\mathcal{L}_{q/\mu} = ||\boldsymbol{\mu} - \boldsymbol{\mu}_{\text{ref}}||^2_{\mathbf{\Sigma}_\mu^{-1}} + ||\boldsymbol{q} - \boldsymbol{q}_{\text{ref}}||^2_{\mathbf{\Sigma}_q^{-1}} + ||\mathbf{w}||^2_{\mathbf{K}_{MM}}. \tag{31}$$

Here, separate regularization parameters can be used for the different properties. This allows weighting the properties relative to each other.

Note that for simplicity, all expression provided herein assume a single kQEq problem with $N$ atoms (*i.e.* one simulation cell or molecule). In practice, models are trained on multiple systems simultaneously using blocked matrices and concatenated vectors, which then naturally allows the use of systems with varying numbers of atoms $N$ in the training set.

**Training on Energies:** As discussed in the introduction, one of the main motivations for developing ML-based charge equilibration models is the development of interatomic potentials with full long-range electrostatics. To this end, training on reference charges (*e.g.* from Hirshfeld partitioning) can yield a reasonable description of long-range interactions. However, population analysis schemes are in general not optimal for this purpose, as the charges usually yield quantitatively incorrect electrostatic properties. More critically, the energy $E_{\text{QEq}}$ is in our experience rather unphysical when only training on charges or dipole moments. This is due to the fact that the energy expression is only a latent quantity in this case (yielding the appropriate charges upon minimization), which bears no relation to the real potential energy surface. As a consequence, the on-site energy contributions can be large and overly sensitive to small geometric changes, making $E_{\mathrm{QEq}}$ a poor basis for an interatomic potential.

One way to overcome this issue is to simply ignore the site-energy term in the interatomic potential. This is the approach taken in the 4GHDNNPs mentioned above.[15] However, this has the downside that the corresponding potentials are not self-consistent, in the sense that their charges do not minimize the energy. The other alternative is to explicitly include energies in the loss function, so that $E_{\mathrm{QEq}}$ takes physical information about the potential energy surface into account.

Unfortunately, fitting energies is not entirely straightforward within the kQEq framework. This is because $E_{\mathrm{QEq}}$ is not linear in the charges, which means that a closed-form solution for the optimal regression weights does not exist. However, for a given set of charges, the energy *is* linear in the electronegativities. `q-pac` therefore includes a form of fixed-point iteration to obtain accurate self-consistent energy models.

Specifically, we use an arbitrary set of initial charges $\mathbf{q}^0$ (*e.g.* from Hirshfeld partitioning) and train electronegativities $\boldsymbol{\chi}^1$ that yield optimal energies for these charges. Subsequently, we predict the self-consistent charges $\mathbf{q}^1$ corresponding to $\boldsymbol{\chi}^1$. In general, there is a large difference between $\mathbf{q}^0$ and $\mathbf{q}^1$, so that the self-consistent energies of this kQEq model will be rather inaccurate. However, iteratively restarting this process usually yields significant improvements, in that the self-consistent charges $\mathbf{q}^t$ corresponding to $\boldsymbol{\chi}^t$ quickly converge towards the ones used to fit the energies ($\mathbf{q}^{t-1}$). In pathological cases, the loss function can be expanded to include a bias towards the charges from the previous iteration, further aiding convergence. Here, a practical approach is to begin training on energies alone until the energy fitting error starts to increase. The charge bias can then be added with an initially large regularization parameter $\sigma_{\mathrm{q}}^2 = 0.01$ e (corresponding to a small weight of charges in the loss function), which is subsequently decreased by a factor of 0.5 at each iteration.

By monitoring the energy fitting error and the charge differences between two iterations, optimal regression weights for a kQEq interatomic potential can be selected. The typical convergence of charges and energies in this process is illustrated for a set of ZnO nanoparticles in Fig. 2 (see below for details on the dataset). This shows that even without the charge bias, energies and charges converge well, with the differences between self-consistent and fitting charges being below $10^{-3}$ electron charges.

**Atomic Forces:** To apply interatomic potentials in molecular dynamics (MD) simulations

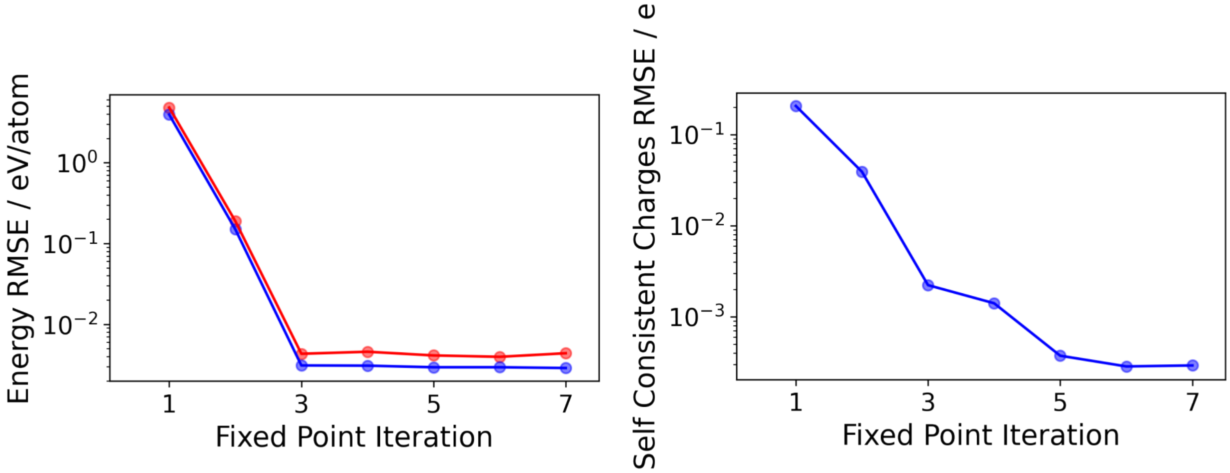

FIG. 2. RMSE of training set (blue) and validation set (red) energies as a function of fixed point iterations, for a set of ZnO nanoparticles (left). RMSE between self-consistent charges and charges from the previous fixed point iteration (right). See Section III for a description of the ZnO nanoparticles and the employed training and validation sets.

and geometry relaxations it is essential to efficiently obtain energy derivatives with respect to atomic positions $\mathbf{r}_j$. While the derivative of Eq. 6 with respect to $\mathbf{r}_j$ is straightforward to compute, a complication arises because both the charges $\mathbf{q}$ and the electronegativities $\boldsymbol{\chi}$ also depend on $\mathbf{r}_j$. Here, the use of self-consistent charges is beneficial, because by definition:

$$\frac{\partial E_{\mathrm{QEq}}}{\partial q_i} = 0 \tag{32}$$

Consequently, the force on atom $j$ can be expressed as

$$\mathbf{F}_j = -\sum_{i=1}^{N}\left(q_i \frac{\partial \chi_i}{\partial \mathbf{r_j}}\right) + \sum_{i>j}^{N} q_i q_j \frac{\partial V_{ij}}{\partial \mathbf{r_j}}, \tag{33}$$

with

$$V_{ij} = \frac{\mathrm{erf}\left(\frac{r_{ij}}{\sqrt{2}\gamma_{ij}}\right)}{r_{ij}}. \tag{34}$$

Here, the first term describes the force caused by the response of the atomic electronegativities to changes in $\mathbf{r}_j$. According to Eq. 21, this term only requires taking the derivatives of the kernel function and the SOAP vectors, the latter of which are obtained through `Dscribe`.

Meanwhile, the second term is simply the derivative of the shielded Coulomb energy, which reads:

$$\frac{\partial V_{ij}}{\partial \mathbf{r_j}} = \frac{\sqrt{2}\gamma_{ij}\exp\left(-\frac{r_{ij}^2}{2\gamma_{ij}^2}\right) - \sqrt{\pi}\mathrm{erf}\left(\frac{r_{ij}}{\sqrt{2}\gamma_{ij}}\right)}{\sqrt{\pi}r_{ij}^3} \tag{35}$$

**Technical Aspects:** q-pac is implemented as an object-oriented library using Python 3.9. It heavily relies on numpy[40] for array operations and linear algebra, ase[41,42] for representing structural data and running atomistic simulations, and Dscribe[37] for calculating SOAP vectors and their derivatives. The Ewald summation portion of the code is written in C++. C++ types are exposed to Python via pybind11[43].

## III. RESULTS

**Effect of Sparsification**: The main algorithmic advance in q-pac relative to the original kQEq implementation is the use of sparse GPR instead of KRR for predicting electronegativities. In order to demonstrate the benefit of this change, a series of dipole moment prediction models were trained, similar to Ref. 34. Specifically, 35,000 randomly selected molecules from the QM9 database were used, spanning a variety of small organic molecules containing C, H, O, N and F.[44] The corresponding reference dipole moments were computed at the PBE0/def2-TZVP level using ORCA.[45–47] From this, 1,000 molecules each were randomly drawn as validation and test sets, while differently sized training sets were randomly drawn from the remaining 33,000 molecules.

As described in the previous section, multiple hyperparameters need to be defined for any kQEq model. Following Ref. 34, the non-classical atomic hardness $J_i$ was set to 0 for all elements. Atomic radii $\alpha_i$ for all elements are tabulated in the original QEq paper.[30] In our previous work we found it beneficial to scale these, since they are not necessarily ideal for the Gaussian charge distributions used in q-pac. In this paper, all radii are globally scaled by $\frac{1}{\sqrt{2}}$, which yielded robust models in all cases we considered.

The regularization strength $\sigma_\mu$ was optimized for each training set using grid search on the validation set error. The main SOAP hyperparameter to be chosen is the cut-off radius $r_{\text{cut}}$, which was set to 4.4 Å. The remaining SOAP hyperparameters are discussed in the Supplementary Information, together with results for smaller values of $r_{\text{cut}}$.

To establish the accuracy of the sparse GPR approach, Fig. 3 shows the mean absolute error (MAE) in predicted dipole moments for different training sets as a function of the number of sparse points $M$ (per element). For comparison, the dashed lines in these figures show results for the corresponding full GPR models, where $M$ includes all chemical environments in the training set. This is equivalent to the KRR models used in the original kQEq paper.[34]

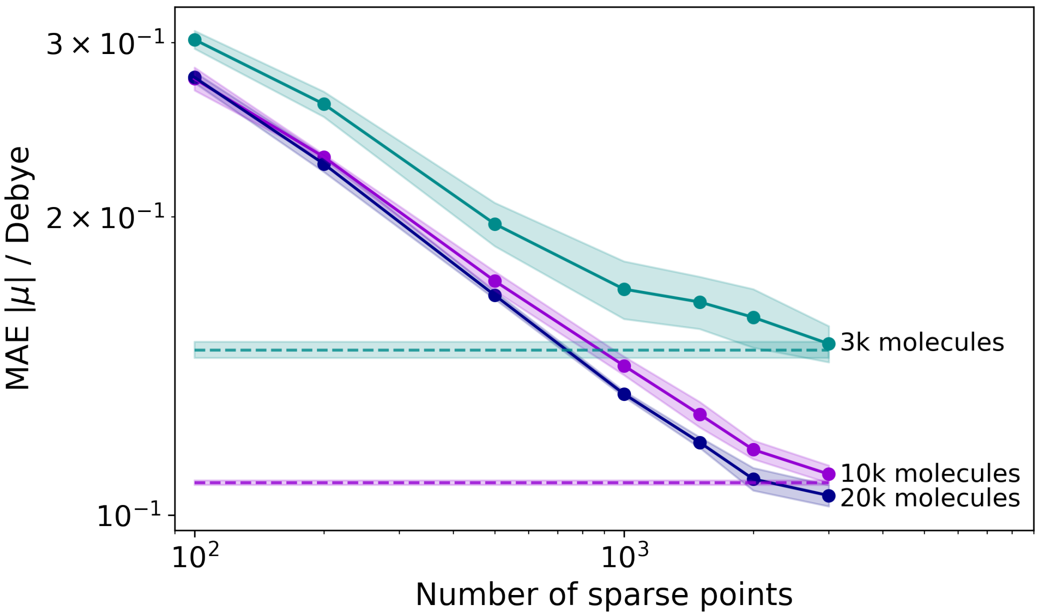

FIG. 3. Comparison of sparse (full line) and full (dashed line) kQEq models for training sets of 3,000 (cyan), 10,000 (purple) and 20,000 (blue) molecules. Three different randomized training sets are used and averaged for each point.

For training sets of 3,000 and 10,000 molecules, the accuracy of the full GPR is reached with $M$=3,000 (0.12 and 0.1 D, respectively). Note that for the training set of 20,000 molecules the full model was not computed due to prohibitively high memory requirements. With sparse GPR this is not an issue, however, allowing even lower MAEs.

In Fig. 4 the evolution of test set errors and computational costs for training and prediction with the number of training molecules is shown. Here it should be emphasized that the number of sparse points is given in terms of chemical environments, whereas the training set size is given in terms of the number of molecules. Depending on the training set size, sparse models with 1,000 and 3,000 sparse points are equally accurate as the full models, with the $M$=1,000 models deviating from the full results for training sets larger than 1,000 molecules. As mentioned above, the full models become computationally prohibitive for the largest training set of 20,000 molecules.

This can also be seen from the timings in Fig. 4, which clearly show that the complete training process (including evaluation of the validation set for hyperparameter tuning) scales much more steeply with the training set size for the full model. This is in line with the expected asymptotic scalings of $\mathcal{O}(N^3)$ and $\mathcal{O}(N)$ for the full and sparse models, respectively, though the sparse models are not yet in the linear regime in this plot. In concrete terms, training and validation of the full models took on average 2.1 hours (see Supplementary Information for hardware details) for 10,000 training molecules. In comparison, the $M$=3,000 model was six times faster, reaching nearly identical accuracy in 0.35 hours. With 1,000 sparse points per elements, the same results were produced in 0.24 hours.

The difference between sparse and full GPR models is even more striking when looking at the prediction times. Here, the sparse models scale as $\mathcal{O}(1)$, while the full models scale as $\mathcal{O}(N)$. This clearly has significant implications for models with large training sets and/or a large number of required predictions, where sparse kQEq models can potentially be one to two orders of magnitude more efficient. Beyond this substantial acceleration, the memory requirements of training sparse models are also much smaller.

**Interatomic Potentials for Isolated Systems:** To illustrate the capabilities of kQEq for fitting potential energy surfaces (PES) of ionic materials, we developed an ML potential for a set of ZnO nanoparticles taken from the global optimization study of Chen and Dixon.[48] Specifically, a set of 98 low-energy structures of sizes between 62 and 264 zinc and oxygen atoms was used. Reference energies and Hirshfeld charges were computed with FHI-Aims at the PBE level using *tight* basis sets and integration settings.[49,50]

Note that the choice of a predominantely ionic material like ZnO allows fitting accurate interatomic potentials using kQEq alone. In general, many materials and molecules display a significant degree of covalent character, which cannot adequately be described within a charge equilibration framework. In this case, the kQEq model should be combined with a regular local interatomic potential, as is done in 4GHDNNPs.

The original dataset of Chen and Dixon exclusively consists of locally relaxed structures. In order to also test kQEq potentials on non-equilibrium structures, active learning was used to augment the dataset.[4,51] Specifically, the initial kQEq interatomic potential (trained on the relaxed structures) was used to generate new configurations via MD simulations. From these, a diverse subset was selected via CUR decomposition, evaluated by the reference DFT method, and added to the training set. This active learning loop was repeated until the energy RMSE on newly generated structures was converged.

The corresponding MD simulations were run through the Atomic Simulation Environment (ASE)[41,42] calculator implemented in `q-pac`. Langevin dynamics were performed at 300 K with a 0.5 fs time step and a friction coefficient of 0.01. Before each production run,

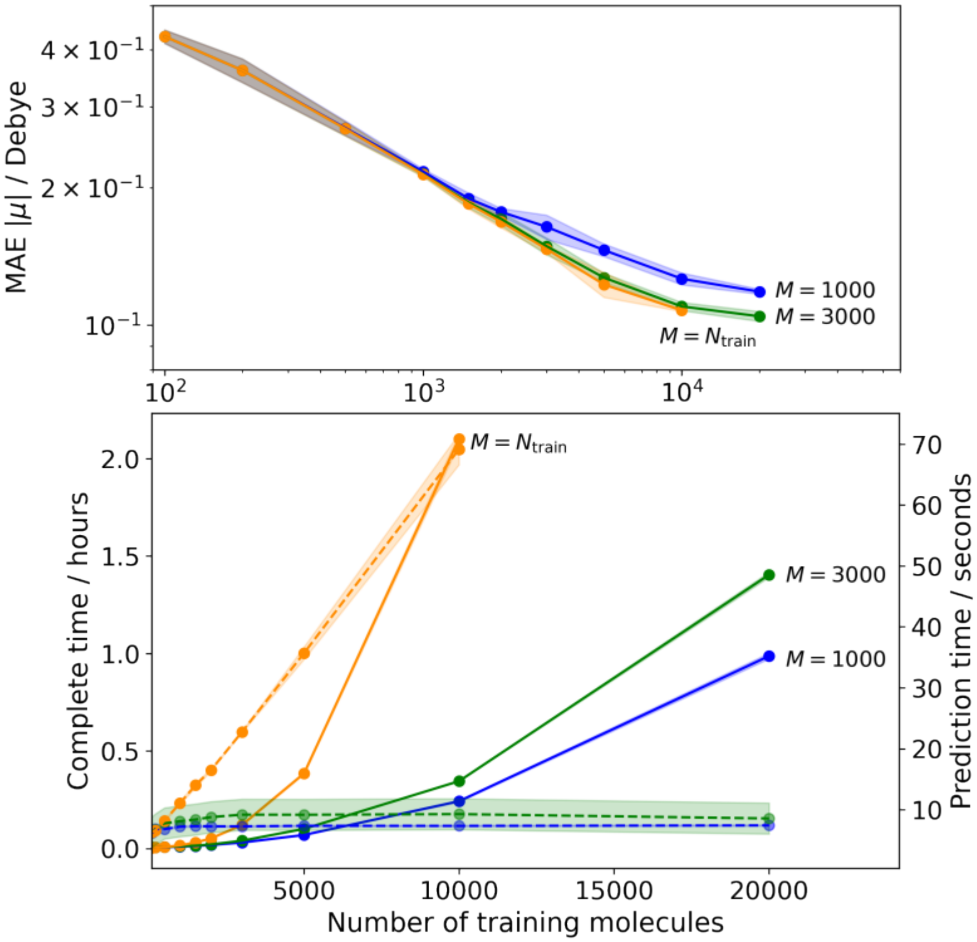

FIG. 4. Top: Learning curves for dipole moment prediction using sparse ($M$ = 1,000 and $M$ = 3,000) and full ($M = N_{\mathrm{train}}$) GPR. Bottom: Time needed for a full training cycle including CUR decomposition (where applicable), model training and validation (solid lines, left y-axis) and timings for 1,000 dipole moment predictions (dashed lines, right y-axis). These calculations were performed on nodes with 2 Intel(R) Xeon(R) Platinum 8360Y CPUs (2.4 GHz) and 2048 GB RAM.

structures were reoptimized with the BFGS algorithm, followed by a 1 ps equilibration run. To account for the increasing accuracy of the interatomic potentials as a function of the active learning iterations, the length of the production runs was incrementally increased. Specifically, the initial simulation time was set to 0.2 ps and increased by a factor of two in each following iteration. Accordingly, the number of configurations added to the training set was also increased in each iteration, starting with 50 configurations.

In terms of model hyperparameters, the regularization for the energy term was set to $\sigma_E^2$=0.01 eV throughout. As discussed in the methods section, a charge bias term was added

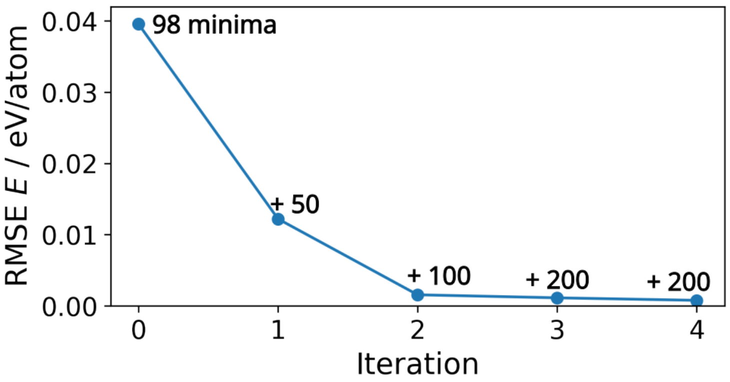

FIG. 5. Active learning potential for ZnO nanoparticles. Shown is the RMSE of predicted energies for new MD configurations, generated with the potential at each active learning iteration. For each iteration, the number of (added) training configurations is shown. The final RMSE was computed for 200 unseen configurations.

in later fixed point iterations to ensure convergence, though this was usually not necessary (see Fig. 2). The hardness parameters $J_i$ were set to 27.21 eV (1.0 Ha) for Zn and O, and the SOAP cutoff was set to 3.0 Å. A full table of hyperparameters is provided in the SI. Note that for this proof-of-principle application no hyperparameter optimization was performed. It is therefore likely that even better performance could be achieved in principle.

Results of the active learning iterations are shown in Fig. 5. Here, the energy RMSE for new MD configurations is shown for each iteration, along with the number of configurations added to the training set. The RMSE quickly drops from 39.6 meV/atom for the initial model to 1 meV/atom in iteration 2 and 0.8 meV/atom in the final iteration.

Beyond this good energetic accuracy, it is also of interest to consider the charge distributions that are learned by this potential. In Fig. 6 (top), the correlation between Hirshfeld and kQEq charges is shown. This reveals that kQEq charges are somewhat larger than Hirshfeld ones, with average charges of ±0.54 and ±0.37, respectively. This is consistent with the known tendency of Hirshfeld charges to underestimate charge transfer in polar systems, which is related to the use of neutral isolated atom densities to define the partitioning.[52,53]

Notably, the variation of the charges displays an inverse correlation between kQEq and Hirshfeld, in particular for the oxygen atoms. Here, the least negative atoms in kQEq are the most negative in Hirshfeld and *vice versa*. This can be understood by considering the individual cases displayed in the lower part of Fig. 6, which reveals that Hirshfeld charges

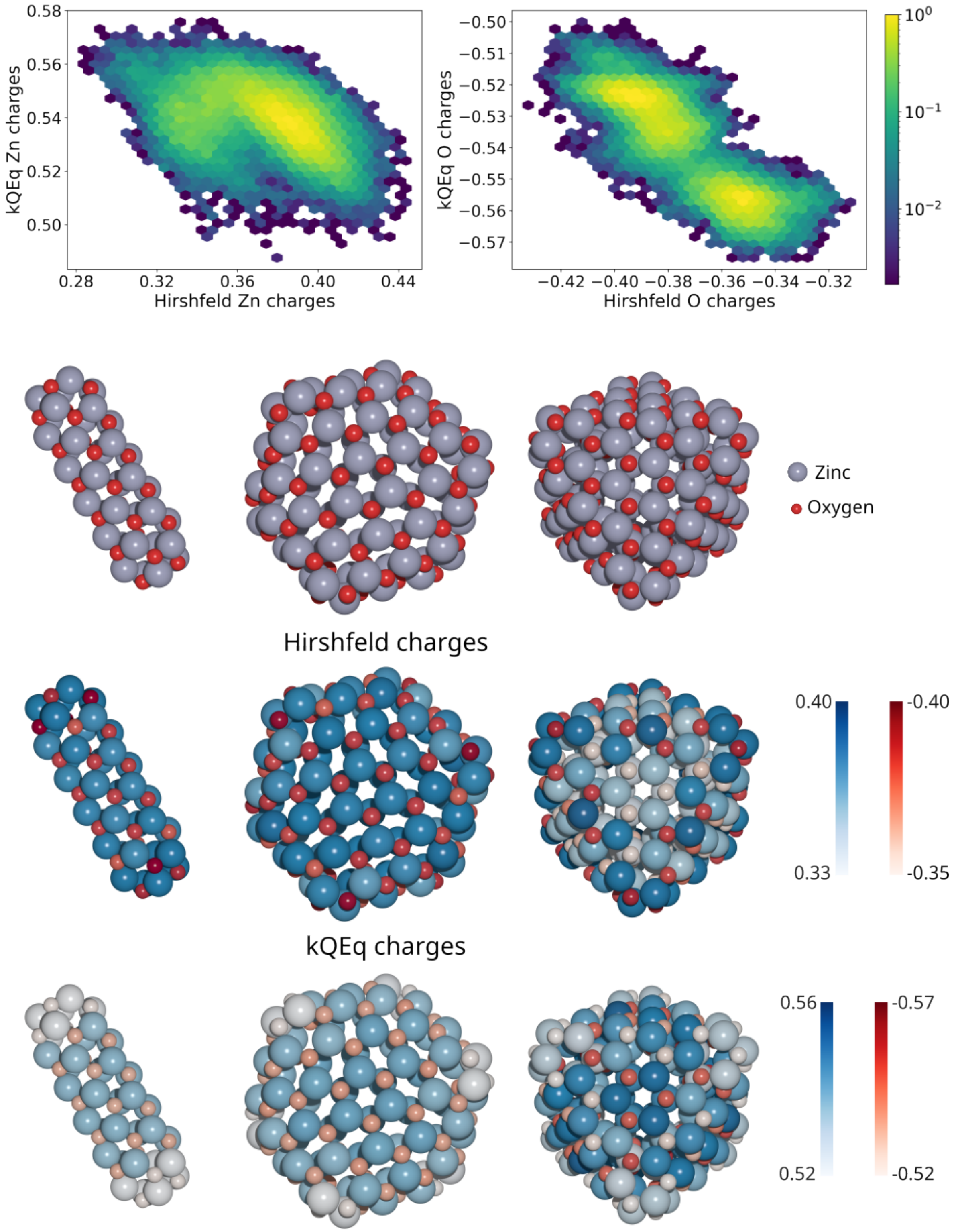

FIG. 6. Top: Correlation between Hirshfeld and kQEq charges for ZnO nanoparticles. Bottom: Comparison of charges for three representative structures, namely a nanotube, a hollow sphere and a dense particle. Note that different color bars are used for Hirshfeld and kQEq charges since the latter are significantly larger in absolute terms.

tend to be more extreme for undercoordinated edge and corner atoms, while kQEq charges are less extreme for these atoms. This is again consistent with the way the respective charges are calculated. In Hirshfeld population analysis, the electron density is spatially partitioned according to a stockholder scheme where the electron densities of isolated neutral atoms are used to define the respective weights. Here, lower coordination environments mean that the density is partitioned between fewer partners. Meanwhile, kQEq charges minimize an electrostatic energy expression. Here, more extreme charges can be stabilized through electrostatic interactions in higher coordination environments.

Overall, partial charges are in general somewhat arbitrary, so that no strong conclusions in favor of either model can be drawn from the charges alone. Nonetheless, it is notable that there are qualitative differences between them. Importantly though, the kQEq charges are optimized for producing an interatomic potential, while the Hirshfeld charges are not.

**Interatomic Potentials for Periodic Systems:** As described in the methods section, `q-pac` also allows developing models for periodic systems. To demonstrate this, we fit an interatomic potential for a range of bulk structures with the stoichiometries ZnO and $ZnO_2$.

An initial set of crystals was obtained from the Materials Project[54] (16 for ZnO and 9 for $ZnO_2$, see SI). This set was augmented by creating random neutral vacancy defect pairs (*i.e.* removing one O and one Zn atom), in different supercells of each crystal, yielding 20 configurations for each supercell. Here, the supercells were used to sample different defect densities and were chosen in order to still allow reference DFT calculations for all cells (*i.e.* containing less than 400 atoms, see SI for details). Additionally, non-equilibrium structures were generated by randomly perturbing the atomic positions with Gaussian noise, yielding 20 additional structures per supercell (half of them perturbed with $\sigma$=0.05 Å, the other half with $\sigma$=0.1 Å). This led to a total of 1,025 structures with supercells ranging from 52 to 384 atoms, for which reference calculations were performed at the PBE level using FHI-Aims with *light* basis set and integration settings.

In order to train the kQEq model, a training set was generated from this data by randomly drawing 150 configurations each for pristine and defected ZnO structures, respectively. Similarly, 80 $ZnO_2$ configurations each were drawn from the pristine and defected sets. The remaining 606 structures were used as an unseen test set. The same hyperparameters as for the non-periodic structures were used for the kQEq and SOAP settings.

The corresponding model displays an RMSE of 4.1 meV/atom on the test set. This is

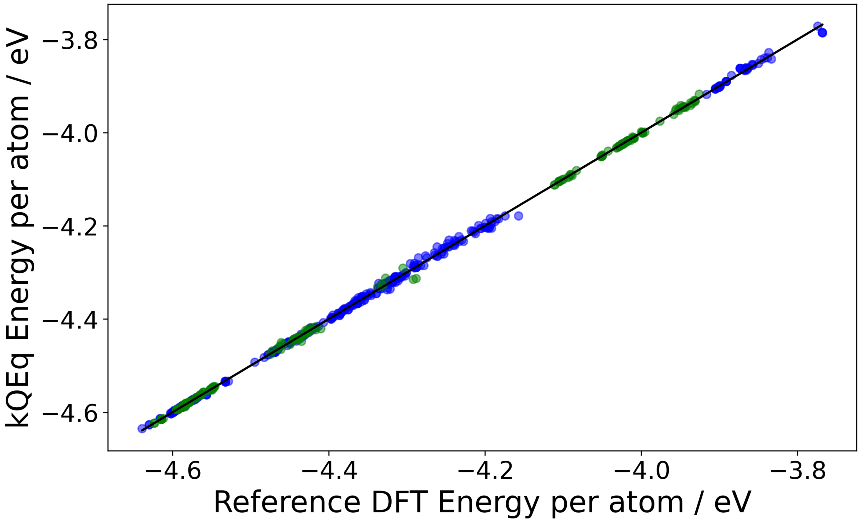

FIG. 7. Parity plot of kQEq predicted energies and reference DFT calculations for a test set of periodic ZnO (blue) and $ZnO_2$ (green) structures. The root mean squared error of the model is 4.1 meV/atom.

quite satisfactory, given that the test set covers an energy span of nearly 1 eV/atom (see Fig. 7). Importantly, the potential is able to fit ZnO and $ZnO_2$ on the same footing, since it is able to describe different oxidation states of Zn through the environment dependent electronegativities.

This is also reflected in Fig. 8, which shows the correlation between Hirshfeld and kQEq charges for the test set. Both approaches yield distinct clusters for the ZnO and $ZnO_2$ charges, with the charges again being larger in magnitude for kQEq (*e.g* ±0.55 versus ±0.35 for ZnO). Notably, the Hirshfeld oxygen charges display a strikingly large variation for $ZnO_2$ and are even positive in some cases, whereas the corresponding kQEq charges are consistently negative and display much smaller variance.

To illustrate the corresponding charge distributions, three simulation cells for a tetraauricupride-structured ZnO polymorph (MP-ID 13161) are shown in Fig. 8. As a reference, Hirshfeld and kQEq charge distributions for the relaxed supercell are shown in the left column. In the center, a rattled configuration is shown. kQEq and Hirshfeld charges display a qualitatively similar response to this perturbation. Finally, the right frame shows a structure with a vacancy pair. Here, an O atom is missing on the bottom right and a Zn atom is missing on the bottom left. Interestingly, the response to the O vacancy is almost identical

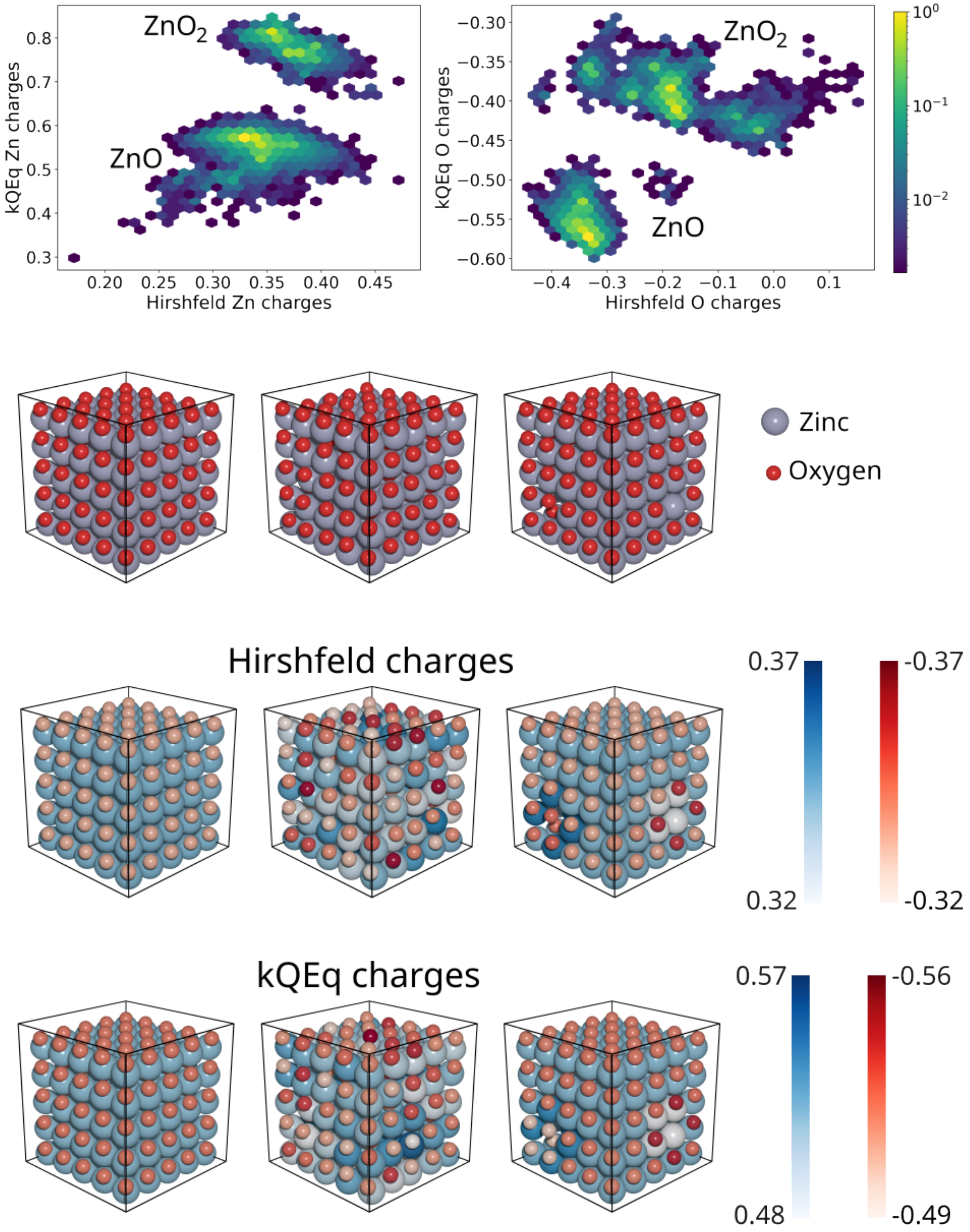

FIG. 8. Top: Correlation between Hirshfeld and kQEq charges for ZnO and $ZnO_2$ bulk structures. Bottom: Comparison of charges for a relaxed ZnO structure, randomly rattled structure and structure with a vacancy pair (MP ID 13161, 5 × 5 × 5 supercell). Note that different color bars are used for Hirshfeld and kQEq charges since the latter are significantly larger in absolute terms.

between the kQEq and Hirshfeld charges, with the adjacent O atoms being more negative and the adjacent Zn atoms being less positive, consistent with an overall charge neutral defect. Meanwhile, the response for the Zn vacancy is similar for the adjacent Zn atoms (which become more positive) but different for the adjacent oxygen atoms. Here, the corresponding Hirshfeld charges are slightly more negative than average, while the kQEq charges are slightly less negative. Nevertheless, the agreement between these methods in terms of the electronic localization of the defect is rather good, given that both methods are based on very different premises.

## IV. CONCLUSIONS

In this paper, we have introduced the `q-pac` package, which provides an efficient and general framework for fitting kQEq models. This is achieved through a new sparse GPR formulation of the kQEq method and the implementation of additional and generalizable fitting targets. Importantly, this allows fitting kQEq interatomic potentials for the first time, using a fixed point iteration algorithm.

While the showcased applications demonstrate the functionality and accuracy of this approach, pure kQEq potentials are (much like the CENT method)[27] limited to predominantly ionic materials like ZnO. In future work, we will explore hybrid potentials that combine short-ranged local interatomic potentials like GAP with kQEq, in order to obtain generally applicable potentials with full long-range interactions.

The modularity of `q-pac` will also allow developing ML charge equilibration models beyond the simple QEq energy function. This may become necessary for properly modelling the response of polarizable systems to external fields, where conventional QEq is know to be inadequate.[55] However, there are some indications that the much larger flexibility of kQEq mitigates these problems to a large extent.[34] Either way, `q-pac` provides the ideal testing ground for addressing such pathologies both in terms of the physics of charge equilibration and in terms of more advanced ML approaches.

**Data and Code Availability:** The code and data for this paper are publicly available at `https://gitlab.com/jmargraf/kqeq`

## REFERENCES

[1] Behler, J. Four generations of high-dimensional neural network potentials. *Chem. Rev.* **121**, 10037–10072 (2021). URL `http://doi.org/10.1021/acs.chemrev.0c00868`.

[2] Margraf, J. T., Jung, H., Scheurer, C. & Reuter, K. Exploring catalytic reaction networks with machine learning. *Nature Catalysis* **6**, 112–121 (2023). URL `https://doi.org/10.1038/s41929-022-00896-y`.

[3] Unke, O. T. *et al.* Machine learning force fields. *Chem. Rev.* **121**, 10142–10186 (2021). URL `http://doi.org/10.1021/acs.chemrev.0c01111`.

[4] Deringer, V. L. *et al.* Gaussian process regression for materials and molecules. *Chem. Rev.* **121**, 10073–10141 (2021). URL `http://doi.org/10.1021/acs.chemrev.1c00022`.

[5] Margraf, J. Science-driven atomistic machine learning. *Angew. Chem. Int. Ed.* e202219170 (2023). URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/anie.202219170`.

[6] Behler, J. & Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* **98**, 583 (2007). URL `http://doi.org/10.1103/PhysRevLett.98.146401`.

[7] Bartók, A. P., Payne, M. C., Kondor, R. & Csányi, G. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Phys. Rev. Lett.* **104** (2010). URL `http://doi.org/10.1103/PhysRevLett.104.136403`.

[8] Stocker, S., Gasteiger, J., Becker, F., Günnemann, S. & Margraf, J. T. How robust are modern graph neural network potentials in long and hot molecular dynamics simulations? *Mach. Learn.: Sci. Technol.* **3**, 045010 (2022). URL `http://doi.org/10.1088/2632-2153/ac9955`.

[9] Kapil, V. *et al.* The first-principles phase diagram of monolayer nanoconfined water. *Nature* **609**, 512–516 (2022). URL `http://doi.org/10.1038/s41586-022-05036-x`.

[10] Morawietz, T., Singraber, A., Dellago, C. & Behler, J. How van der waals interactions determine the unique properties of water. *Proc. Natl. Acad. Sci. U.S.A.* **113**, 8368–8373 (2016). URL `http://doi.org/10.1073/pnas.1602375113`.

[11] Deringer, V. L. *et al.* Origins of structural and electronic transitions in disordered silicon. *Nature* **589**, 59–64 (2021). URL `http://doi.org/10.1038/s41586-020-03072-z`.

[12] Herbold, M. & Behler, J. A hessian-based assessment of atomic forces for training machine learning interatomic potentials. *J. Chem. Phys.* **156**, 114106 (2022). URL `http://doi.`

org/10.1063/5.0082952.

[13] Deringer, V. L. & Csányi, G. Machine learning based interatomic potential for amorphous carbon. *Phys. Rev. B* **95** (2017). URL http://doi.org/10.1103/PhysRevB.95.094203.

[14] Staacke, C. G. *et al.* On the role of long-range electrostatics in machine-learned interatomic potentials for complex battery materials. *ACS Appl. Energy Mater.* **4**, 12562–12569 (2021). URL http://doi.org/10.1021/acsaem.1c02363.

[15] Ko, T. W., Finkler, J. A., Goedecker, S. & Behler, J. A Fourth-Generation High-Dimensional Neural Network Potential with Accurate Electrostatics Including Non-local Charge Transfer. *Nat. Commun.* **12**, 398 (2021).

[16] Wengert, S., Csányi, G., Reuter, K. & Margraf, J. T. Data-efficient machine learning for molecular crystal structure prediction. *Chem. Sci.* **12**, 4536–4546 (2021). URL http://doi.org/10.1039/D0SC05765G.

[17] Wengert, S., Csányi, G., Reuter, K. & Margraf, J. T. A hybrid machine learning approach for structure stability prediction in molecular co-crystal screenings. *J. Chem. Theory Comput.* **18**, 4586–4593 (2022). URL http://doi.org/10.1021/acs.jctc.2c00343.

[18] Jung, H. *et al.* Size-extensive molecular machine learning with global representations. *ChemSystemsChem* **2**, 659 (2020). URL http://doi.org/10.1002/syst.201900052.

[19] Chmiela, S. *et al.* Accurate global machine learning force fields for molecules with hundreds of atoms. *Sci. Adv.* **9**, 1875 (2023). URL http://doi.org/10.1126/sciadv.adf0873.

[20] Grisafi, A. & Ceriotti, M. Incorporating long-range physics in atomic-scale machine learning. *J. Chem. Phys.* **151**, 204105 (2019). URL http://doi.org/10.1063/1.5128375.

[21] Ramakrishnan, R., Dral, P. O., Rupp, M. & Von lilienfeld, O. A. Big data meets quantum chemistry approximations: The $\delta$-machine learning approach. *J. Chem. Theory Comput.* **11**, 2087–2096 (2015). URL http://doi.org/10.1021/acs.jctc.5b00099.

[22] Fan, G., Mcsloy, A., Aradi, B., Yam, C.-Y. & Frauenheim, T. Obtaining electronic properties of molecules through combining density functional tight binding with machine learning. *J. Phys. Chem. Lett.* **13**, 10132–10139 (2022). URL http://doi.org/10.1021/acs.jpclett.2c02586.

[23] Westermayr, J., Chaudhuri, S., Jeindl, A., Hofmann, O. T. & Maurer, R. J. Long-range dispersion-inclusive machine learning potentials for structure search and optimization of hybrid organic–inorganic interfaces. *Digital Discovery* **1**, 463–475 (2022). URL http://doi.org/10.1039/D2DD00016D.

[24]Schütt, K. T., Arbabzadah, F., Chmiela, S., Müller, K. R. & Tkatchenko, A. Quantum-chemical insights from deep tensor neural networks. *Nat Commun* **8**, 190 (2017). URL `http://doi.org/10.1038/ncomms13890`.

[25]Artrith, N., Morawietz, T. & Behler, J. High-dimensional neural-network potentials for multicomponent systems: Applications to zinc oxide. *Phys. Rev. B* **83** (2011). URL `http://doi.org/10.1103/PhysRevB.83.153101`.

[26]Morawietz, T., Sharma, V. & Behler, J. A neural network potential-energy surface for the water dimer based on environment-dependent atomic energies and charges. *J. Chem. Phys.* **136**, 064103 (2012). URL `http://doi.org/10.1063/1.3682557`.

[27]Ghasemi, S. A., Hofstetter, A., Saha, S. & Goedecker, S. Interatomic potentials for ionic systems with density functional accuracy based on charge densities obtained by a neural network. *Phys. Rev. B* **92** (2015). URL `http://doi.org/10.1103/PhysRevB.92.045131`.

[28]Faraji, S. *et al.* High accuracy and transferability of a neural network potential through charge equilibration for calcium fluoride. *Phys. Rev. B* **95**, 1041 (2017). URL `http://doi.org/10.1103/PhysRevB.95.104105`.

[29]Khajehpasha, E. R., Finkler, J. A., Kühne, T. D. & Ghasemi, S. A. Cent2: Improved charge equilibration via neural network technique. *Phys. Rev. B* **105**, 1 (2022). URL `http://doi.org/10.1103/PhysRevB.105.144106`.

[30]Rappe, A. K. & Goddard, W. A. I. Charge equilibration for molecular dynamics simulations. *The Journal of Physical Chemistry* **95**, 3358–3363 (1991). URL `https://doi.org/10.1021/j100161a070`. https://doi.org/10.1021/j100161a070.

[31]Ko, T. W., Finkler, J. A., Goedecker, S. & Behler, J. General-purpose machine learning potentials capturing nonlocal charge transfer. *Acc. Chem. Res.* **54**, 808–817 (2021). URL `http://doi.org/10.1021/acs.accounts.0c00689`.

[32]Hirshfeld, F. L. Bonded-atom fragments for describing molecular charge densities. *Theoret. Chim. Acta* **44**, 129–138 (1977). URL `http://doi.org/10.1007/BF00549096`.

[33]Xie, X., Persson, K. A. & Small, D. W. Incorporating electronic information into machine learning potential energy surfaces via approaching the ground-state electronic energy as a function of atom-based electronic populations. *J. Chem. Theory Comput.* **16**, 4256–4270 (2020). URL `http://doi.org/10.1021/acs.jctc.0c00217`.

[34]Staacke, C. G. *et al.* Kernel charge equilibration: efficient and accurate prediction of molecular dipole moments with a machine-learning enhanced electron density model. *Machine*

*Learning: Science and Technology* **3**, 015032 (2022).

[35] Ramachandran, S., Lenz, T. G., Skiff, W. M. & Rappé, A. K. Toward an understanding of zeolite y as a cracking catalyst with the use of periodic charge equilibration. *The Journal of Physical Chemistry* **100**, 5898–5907 (1996). URL `https://doi.org/10.1021/jp952864q`.

[36] Gingrich, T. R. & Wilson, M. On the ewald summation of gaussian charges for the simulation of metallic surfaces. *Chemical Physics Letters* **500**, 178–183 (2010). URL `https://www.sciencedirect.com/science/article/pii/S0009261410013606`.

[37] Himanen, L. *et al.* Dscribe: Library of descriptors for machine learning in materials science. *Computer Physics Communications* **247**, 106949 (2020). URL `https://www.sciencedirect.com/science/article/pii/S0010465519303042`.

[38] Mahoney, M. W. & Drineas, P. Cur matrix decompositions for improved data analysis. *PNAS* **106**, 697–702 (2008).

[39] Bartók, A. P., Kondor, R. & Csányi, G. On representing chemical environments. *Phys. Rev. B* **87** (2013). URL `http://doi.org/10.1103/PhysRevB.87.184115`.

[40] Harris, C. R. *et al.* Array programming with NumPy. *Nature* **585**, 357–362 (2020). URL `https://doi.org/10.1038/s41586-020-2649-2`.

[41] Larsen, A. H. *et al.* The atomic simulation environment—a python library for working with atoms. *Journal of Physics: Condensed Matter* **29**, 273002 (2017). URL `http://stacks.iop.org/0953-8984/29/i=27/a=273002`.

[42] Bahn, S. R. & Jacobsen, K. W. An object-oriented scripting interface to a legacy electronic structure code. *Comput. Sci. Eng.* **4**, 56–66 (2002).

[43] Jakob, W., Rhinelander, J. & Moldovan, D. pybind11 – seamless operability between c++11 and python (2017). Https://github.com/pybind/pybind11.

[44] Ramakrishnan, R., Dral, P. O., Rupp, M. & von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data* **1**, 1–7 (2014).

[45] Neese, F. The orca program system. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2**, 73–78 (2012).

[46] Neese, F. Software update: The orca program system—version 5.0. *WIREs Computational Molecular Science* **12**, e1606 (2022). URL `https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wcms.1606`. https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wcms.1606.

[47] Perdew, J. P., Ernzerhof, M. & Burke, K. Rationale for mixing exact exchange with

density functional approximations. *J. Chem. Phys.* **105**, 9982–9985 (1996). URL http://doi.org/10.1063/1.472933.

[48]Chen, M. & Dixon, D. A. Machine-learning approach for the development of structure–energy relationships of zno nanoparticles. *The Journal of Physical Chemistry C* **122**, 18621–18639 (2018). URL https://doi.org/10.1021/acs.jpcc.8b01667. https://doi.org/10.1021/acs.jpcc.8b01667.

[49]Blum, V. *et al.* Ab initio molecular simulations with numeric atom-centered orbitals. *Computer Physics Communications* **180**, 2175–2196 (2009). URL https://www.sciencedirect.com/science/article/pii/S0010465509002033.

[50]Perdew, J. P., Burke, K. & Ernzerhof, M. Generalized gradient approximation made simple. *Phys. Rev. Lett.* **77**, 3865–3868 (1996). URL http://doi.org/10.1103/PhysRevLett.77.3865.

[51]Artrith, N. & Behler, J. High-dimensional neural network potentials for metal surfaces: A prototype study for copper. *Phys. Rev. B* **85**, 045439 (2012). URL https://link.aps.org/doi/10.1103/PhysRevB.85.045439.

[52]Bultinck, P., Van alsenoy, C., Ayers, P. W. & Carbó-dorca, R. Critical analysis and extension of the hirshfeld atoms in molecules. *J. Chem. Phys.* **126**, 144111 (2007). URL http://doi.org/10.1063/1.2715563.

[53]Marenich, A. V., Jerome, S. V., Cramer, C. J. & Truhlar, D. G. Charge model 5: An extension of hirshfeld population analysis for the accurate description of molecular interactions in gaseous and condensed phases. *J. Chem. Theory Comput.* **8**, 527–541 (2012). URL http://doi.org/10.1021/ct200866d.

[54]Jain, A. *et al.* Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL Materials* **1**, 011002 (2013). URL http://doi.org/10.1063/1.4812323.

[55]Koski, J. P. *et al.* Water in an external electric field: Comparing charge distribution methods using reaxff simulations. *J. Chem. Theory Comput.* **18**, 580–594 (2022). URL http://doi.org/10.1021/acs.jctc.1c00975.