Development of a machine-learning–based ionic-force correction model for quantum molecular dynamic simulations of warm dense matter

Joshua P. Hinz, Valentin V. Karasiev*, Suxing X. Hu, Deyan I. Mihaylov

Laboratory for Laser Energetics, University of Rochester,

Rochester, NY 14623-1299

*E-mail: vkarasev@lle.rochester.edu

Abstract

In this work *Δ*-learning is used to map orbital-free density functional theory (OF-DFT) ionic forces to the corresponding Kohn-Sham (KS) DFT ionic forces. The development of the approximate force difference in terms of the ion positions is constructed and serves as a stand in for the ground truth force difference. Descriptor vectors for ion configurations are constructed using all distance between ions in conjunction with an indexing based on a nearest neighbor ranking. It is demonstrated that such a scheme of descriptors can uniquely describe an ionic configuration up to a rotation and reflection when no ambiguity in the nearest neighbor ranking exists. How to handle the case when an ambiguity exists in the nearest neighbor ranking is discussed. As a proof of principle, the model is trained and tested on warm dense hydrogen at temperatures between 1 and 15 eV. Once tested, the model was used to perform molecular dynamic simulations of warm dense hydrogen. The resulting energies and pressures are within 1% and 2% of their respective target KS values.

I. Introduction

In the warm-dense-matter (WDM) regime, *ab initio* molecular-dynamic (MD) simulations have become an important tool in the investigation of material properties. These simulations often rely upon density functional theory (DFT) [1–3] calculations of the electronic ground state. This is due to DFT's ability to balance accuracy with computational cost. However, the standard approach of finite-temperature Kohn–Sham (KS) DFT [2,4] has a computational cost that scales cubically with the number of thermally occupied orbitals [5,6]. This limits KS-DFT based MD (KSMD) runs to only a few thousand steps or less for temperatures approaching (and above) the Fermi Temperature (TF) of the system. As an alternative approach, finite temperature orbital-free (OF) DFT [7, 8, 9] is orders of magnitude faster than KS-DFT at temperatures consistent with the WDM regime [5,6]. Unfortunately, OF-DFT requires an approximate functional for the non-interacting free energy in terms of electron density [5,6]. Even with the best approximate non-interacting free energy functionals today [10, 11], reliable accuracy of OF-DFT is only achieved for temperatures above ~5TF. For WDM simulations with temperatures between ~0.5TF and ~5TF, it would be advantageous to have a method that captures the best of both KS and OF DFT.

Over the past decade, approaches in machine learning (ML) have provided such schemes capable of delivering KS-level accuracy at an OF cost, or faster, for the calculation of ionic forces needed to drive MD simulations [12–18]. These schemes fall into two categories, the first being the direct interpolation of the Born–Oppenheimer potential energy surface (PES) from which analytical derivatives of the model can be taken with respect to the ion positions to generate ionic forces [12,15,19]. The second is a direct construction of force fields [13,14]. The standard approach of both methods is to create a description of the neighboring ions within a predetermined cutoff radius about a given reference ion for which the prediction of the model is being made. The choice of how this description of the local configuration of the reference ion is constructed has important implications for the model’s success and has been an active area of research [12–15,20–22, 23,24]. Furthermore, these ML approaches have been applied to a variety of systems and problems including the simulation of bulk [25,26] and amorphous solids [27], calculations of melting points [28], investigation of solid–liquid interfaces [29], and the prediction of liquid–liquid transitions [30,31], to name a few. For a more-comprehensive list, see the references within the following review articles of Refs. [16–18].

While the direct prediction of energies and forces is prominent throughout the literature, a newly developed ML scheme referred to as *Δ*-learning [32-35] has emerged in recent years. Within *Δ*-learning, a cheaper, less-accurate method is corrected with a ML model in order to produce a more-accurate and expensive target quantity. Recent success of *Δ*-learning models can be seen in the context of references [34,35] that correct DFT energies and forces to produce corresponding coupled cluster quantities.

The goal of this work is to utilize *Δ*-learning to develop a ML model that can correct OF-DFT ionic forces to produce ionic forces that have a KS level accuracy with a near-OF cost. The model will be applied to WDM simulations at temperatures where KS-DFT limits a typical MD simulation to only a few thousand steps and where OF-DFT has not achieved the required accuracy. Moving forward, the paper is laid out as follows: Section II contains the development of the approximate force correction, details of the neural network used in this work and the development of the set of descriptor vectors. Section III contains the description of the reference data. Sections IV provides the results of the model's accuracy. This includes the performance on the test set after training and the resulting performance of the model in the context of MD simulations. Finally, section V provides a summary and outlook on future work with the model.

II. Model development

The base assumption of Δ learning is a correction to an estimator of a quantity of interest is easier to learn than the direct prediction of that quantity [32]. Here, the base estimator will be the OF-DFT ionic force with the quantity of interest being the corresponding KS-DFT ionic force. The force correction model will be tasked with predicting force differences such that during MD simulations an OF calculation of the ionic forces can be corrected to provide an equivalent KS-DFT ionic force as illustrated in Eq. (1):

$$\vec{F}^{\mathrm{KS}} = \vec{F}^{\mathrm{OF}} + \Delta\vec{F}^{\mathrm{ML}} \quad . \tag{1}$$

A two-step approach is taken to construct the force correction. First, the true analytical expression for the target ground truth force difference (see Eq. (2) in the next subsection) will be replaced by an approximate target force difference that is dependent only on the ion positions. Second, a ML framework will be introduced and used to learn a scalar quantity of the new approximate target force difference. This procedure is discussed below beginning with a discussion of the true reference force difference.

A. Approximating the target force difference

All calculations throughout this work are performed with a classical treatment of the ions within the Born-Oppenheimer approximation [36]. Within this context, the KS forces are calculated by first determining the ground state electron density n_0^{KS} and then applying the Hellman-Feynman theorem [37] to the total free energy functional [5]. In the OF-DFT branch, the ionic forces are determined by taking the gradients with respect to the ion positions of the total free energy evaluated at the ground state density n_0^{OF} [38] (see Eq. (7) in Ref. [38] and discussion within Ref. [5]). Together, the ion-ion contribution to the ionic forces cancel and the resulting analytical expression for reference force difference will have the form,

$$\Delta\vec{F}_{\mathrm{i}}^{\mathrm{ref}} = kZ_{\mathrm{i}}q^2 \int d^3r \frac{\Delta n_0(r)}{\left|\vec{r}-\vec{R}_i\right|^3}\left(\vec{r}-\vec{R}_i\right) \quad , \tag{2}$$

where

$$\Delta\vec{F}_{\mathrm{i}}^{\mathrm{ref}} = \vec{F}_{\mathrm{i}}^{\mathrm{KS}} - \vec{F}_{\mathrm{i}}^{\mathrm{OF}} \tag{3}$$

and

$$\Delta n_0(r) = n_0^{\mathrm{KS}}(r) - n_0^{\mathrm{OF}}(r) \quad . \tag{4}$$

Here, $\vec{R}_i$ is the position of the i^{th} ion with nuclear charge Z_{i}: k and q are the electrostatic constant and electronic charge, respectively. Moving forward, the ion for which the forces are calculated/predicted will be referred to as the reference ion and will be strictly indexed by the letter i.

From the target ground truth force difference of Eq. (2) an approximate target force difference is constructed. The first step in this construction is to assume that the long range interactions between the reference ion and the electron density difference are negligibly small beyond a predetermined cutoff radius R_c. Mathematically this amounts to limiting the integration in Eq. (2) to a spherical volume defined by a cutoff radius. Such an approximation has become common place throughout the literature for both force [13,14] and energy [12,15] based models and is needed to help control the computational cost of the model. However, in this work, long range interactions still contribute to the final predicted KS force through the underlying OF force, see Eq. (1); a potential advantage of using a correction model as opposed to directly predicting the KS forces.

The second step towards constructing an approximate target force difference is to divide the volume within the cutoff radius into a set of sub-volumes. The set of sub-volumes is constructed such that it is in a 1-to-1 correspondence with the set of neighboring ions within the cutoff radius. An example is shown in Fig. 1 (a). The purpose of the1-to-1 correspondence is to allow the ion positions to act as grid points for resolving the electron charge difference around the reference ion. Next, the approximation is made to treat the total electronic charge difference within a given sub-volume as if it were located at the position of the corresponding grid point, i.e. neighboring ion position. The resulting approximate force difference for the reference ion has the form,

$$\Delta\vec{F}_{\mathrm{i}} \approx \sum_{j} \frac{\vec{R}_{ij}}{\left|\vec{R}_{ij}\right|^{3}} kq^{2} Z_{\mathrm{i}} \int d^{3}r_{j} \Delta n_{0}\left(r_{j}\right) \quad , \tag{5}$$

where

$$\vec{R}_{ij} = \vec{R}_{j} - \vec{R}_{i} \ . \tag{6}$$

Moving forward, when an ionic position is used to reference a particular sub-volume the corresponding ion will be referred to as a first neighbor (FN).

The final step in the development of an approximate force difference is to eliminate the need to work with the electron densities in Eq. (5). This is done by assuming that the total electronic charge difference within a given sub-volume can be determined by the local ions surrounding that sub-volume. This is illustrated in Fig. 1 (b) where the assumption amounts to replacing an integral over the electron density difference of the blue shaded region with information about the surrounding configuration of ions in green and FN in blue. Moving forward, ions used to describe the electron charge difference of a sub volume will be referred to as second neighbors (SN). During the prediction of a reference ion's force difference all neighboring ions within the cutoff radius will play the role of both a FN and SN (the reference ion will also play the role of a SN for its nearest FNs, and ions just beyond the cutoff radius may play the role of a SN).

The final form of the approximate force difference is

$$\Delta \vec{F}_{\mathrm{i}} \approx \sum_{j} w_{ij} \frac{\vec{R}_{ij}}{\left|\vec{R}_{ij}\right|} \ , \tag{7}$$

where w_{ij} is the magnitude of the contribution of the density difference in the sub-volume referenced by the j^{th} FN to the force difference of the reference ion. Note, that a $\left|\vec{R}_{ij}\right|^2$ term has also been lumped into the definition of the weights, comparison of Eq. (5) and (7). As such, to determine the weight of a particular FN both information about the surrounding SNs and information about how far the FN is from the reference ion must be provided. This was done to add additional flexibility to the model when the ML framework is introduced.

B. Introducing ML into the correction model

Within the context of the force difference of Eq. (7) ML will be introduced. This is done by using a neural network (NN) to predict the FN weights;

$$w_{ij} = NN\left(\vec{d}_{ij}\right) . \tag{8}$$

Here, $\vec{d}_{ij}$ is a vector containing relevant information needed to predict the weight associated with the j^{th} FN as discussed at the end of the previous subsection. The details of how $\vec{d}_{ij}$ is construction can be found in the next subsection. In the case of a NN with a single hidden layer, the FN weights can be written as

$$w_{ij} = NN(\vec{d}_{ij}) = \boldsymbol{W}^{(2)} f(\boldsymbol{W}^{(1)} \vec{d}_{ij} + \vec{\beta}), \tag{9}$$

where matrices $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$ and vector $\vec{\beta}$ consist of adjustable free parameters. The function f is the activation function that acts in an elementwise manner and is set to the rectified linear unit (ReLu) activation function [39] throughout this work.

The resulting predicted ML force difference has the form,

$$\Delta F_{\mathrm{i}}^{\mathrm{ML}} = \sum_{\substack{j \\ |\vec{R}_{ij}|<R_{\mathrm{c}}}} NN(\vec{d}_{ij}) \frac{\vec{R}_{ij}}{|\vec{R}_{ij}|} \quad . \tag{10}$$

It is important to note that an identical NN is used to predict all FN weights. The resulting force correction model will then be analogous to the energy model of Behler and Parinello [12]. Furthermore, the free parameters and correspondingly the force differences of Eq. (10) are optimized by minimizing the cost function

$$C = \frac{1}{2N_S} \sum_{i}^{N_S} \left\{ \left| \Delta \vec{F}_{\mathrm{i}}^{\mathrm{ref}} - \Delta \vec{F}_{\mathrm{i}}^{\mathrm{ML}} \right|^2 + A \exp\left(-a \left| \Delta \vec{F}_{\mathrm{i}}^{\mathrm{ML}} \right|^2 \right) \right\} . \tag{11}$$

Here, N_S is the number of reference ions in the training set. The reference force difference $\Delta \vec{F}_{\mathrm{i}}^{\mathrm{ref}}$ will be calculated according to Eq. (3). That is, both the KS and OF reference forces are calculated in their entirety and then the difference is taken. By using reference values for the exact force differences and not the approximate force differences the NN will be tasked with implicitly learning the boundaries for the sub-volumes associated with the FNs.

When the standard squared loss was used for the reference data set described in the next section, instabilities in the training process would often result in a NN that would predict all force differences as zero. To prevent this, the exponential regularization, second term of Eq.

(11), was added to the cost. The hyperparameters A and α in the regularization term are set such that only force differences smaller than those observed in the training set are penalized. This will leave the global minimum of the square loss based cost unaffected.

For the force correction model to be useful in MD simulations, an uncertainty for each predicted force difference is necessary to ensure that the reliability of the prediction holds [14,16,19,40,41]. To estimate the uncertainty an ensemble approach, similar to the work of Ref. [40], is used. In this work, 15 different NN's will be trained to produce 15 different force-correction models. Each member of this ensemble will be trained according to Eq. (11) with its own training data. Once assembled, the average predicted force difference of the ensemble will replace $\Delta\vec{F}^{\mathrm{ML}}$ of Eq. (1) for a given reference ion. The standard deviation associated with this average will then be used to gauge the level of uncertainty in the prediction.

C. The input descriptor vector

In the construction of the descriptor vectors it is important that the same force difference be predicted for configurations of FNs which differ by a translation or exchange of labels of ions of the same species [12–15,21,23,25]. In the case of rotations about the reference ion, the force difference and FN position vectors, Eq. (6), must transform in the same manner. Through inspection of Eq. (10) it is clear that these properties will be satisfied as long as the descriptor vector for each FN is invariant under translations, rotations and label exchanges. As such, these invariances will be directly incorporated into the following construction of the descriptor vectors.

As discussed at the end of subsection A, the weight of each FN is determined by information about its surrounding configuration of SNs as well as information about how far the particular FN is from the reference ion. To incorporate both pieces of information the descriptor vector will be constructed in two parts. First, a vector $\vec{d}^{T}_{\mathrm{SN}^{(j)},j}$ containing only information about the FN's SNs will be assembled. Second, the distance $\left|\vec{R}_{ij}\right|$ from the FN to the reference ion will be used as an element. The full input vector to the NN will then take the form

$$\vec{d}_{ij} = \left(\vec{d}^{T}_{\mathrm{SN}^{(j)},j} \left|\vec{R}_{ij}\right| \right)^{T} . \tag{12}$$

To construct $\vec{d}^{T}_{\mathrm{SN}^{(j)},j}$ the first n nearest SNs around a given FN will be considered. Each of the SNs will be indexed by their nearest neighbor rank with respect to the corresponding FN. By using the nearest neighbor ranking for the SN's index the indices of the SNs become independent from the label given to the FN and any change in the labels of the SNs must be accompanied by a change in the configuration of SNs. In effect, the nearest neighbor ranking in conjunction with the use of the FN distance in the construction of the full descriptor vector, Eq. (12), automatically builds in the invariance under label exchange. The downside however is the number of SNs n must be set. This can be achieved by performing a series of convergence test similar to how R_C would be set.

The next step in the construction of the descriptor vector is to build in rotational and translational invariance. This will be done simultaneously by using all distances between the ions of the SN configuration. It is of note that using a distribution of these distances will not provide a unique description of the configuration of SNs [11,23]. In an attempt to avoid this issue, the distances themselves will be directly used as the elements of the SN descriptor vector. This requires that the distances be strictly ordered so that the same input vector is formed each time a given SN configuration is encountered. The ordering of the distances can be achieved with the use of the nearest SN ranking. More specifically, any distance which has reference to the FN will come first in the SN descriptor; then distances which reference the 1st SN come next and so on. In the case where two distances reference the same ion, it is the index of the second ion referenced by the distance that determines the ordering. For example, in the case of three SNs the SN descriptor vector takes the form

$$\vec{d}^{T}_{SN^{(j)},j} = \left(\left|\vec{R}_{FN,SN1}\right| \left|\vec{R}_{FN,SN2}\right| \left|\vec{R}_{FN,SN3}\right| \left|\vec{R}_{SN1,SN2}\right| \left|\vec{R}_{SN1,SN3}\right| \left|\vec{R}_{SN2,SN3}\right|\right). \qquad (13)$$

In the case where only one or two SNs are needed, it is clear the SN descriptor uniquely describes the SN configuration. For three or more SNs, with no ambiguity in the nearest SN ranking, uniqueness can also be confirmed as follows.

Due to the translational and rotational invariances of the distances, the SN configuration can be oriented such that the FN is at the origin and the 1st SN is along the z-axis. The 2nd SN can then be rotated into the xz-plane such that it has a positive x value. Note, here it is assumed the 2nd SN does not lie along the z-axis after the initial rotation. In such a case where this assumption is not true, the next SN in the nearest neighbor ranking that does not lie on the z-axis will take the role of the 2nd SN in the following discussion (discussion holds because all distances between ions are used). When the distance between the kth SN, with 3 ≤ k ≤ n, and FN is given, the kth SN must lie on the surface of a sphere that is centered on the FN and has a radius corresponding to the given distance. The same situation occurs when the distance is given between the kth and 1st SN. Now when both the kth SN-FN and kth SN-1st SN distances are used together, the kth SN must live at the intersection of the two spherical surfaces which is a circle about the z-axis. Both the z location and radius of this circle can be determined by the known distances, see [42].

When the distance kth SN–2nd SN is added to the two previous distances, the circle of possible positions for the kth SN is reduced to two possible points. This is because a point in the xz-plane (not on the z-axis) is equidistant from at most two possible points on a circle about the z-axis [42]. Moreover, these two points will be reflections of one another about the xz-plane. More generally, the two points on the circle that are equidistant to an arbitrary point in space (not at the circle's center) will be reflections of one another about the plane with a normal vector in the direction $\vec{r}x\vec{n}_c$; where $\vec{r}$ is the position of the arbitrary point and $\vec{n}_c$ is the normal vector of the plane containing the circle. Since two points cannot be reflections of one another about two different planes in three dimensions, when the distance kth SN-mth SN is added to the six distances that reduce the kth and mth SN to two possible locations each (mth SN is assumed to not be in the xz-plane), the four possible configurations of the kth and mth SN are reduced to two unique configurations and their corresponding reflections about the xz-plane.

By the symmetry argument used above, if the distances between all SNs located outside of the xz-plane are used in conjunction with the distances to the FN, 1st SN and 2nd SN for each of these SNs, then the configuration of these SNs is defined uniquely up to a reflection about the xz-plane. For any SN that lives in the xz-plane the distances to the FN, 1st SN and 2nd SN are sufficient to determine that SN's position. Therefore, when all distances between SNs and SNs-FNs are used, the full configuration of SNs will be uniquely defined up to a rotation and reflection. It is however likely that using all distances between ions to form the input vector leads to redundant information being passed to the NN. This might be alleviated by using the distance to the FN and 1st SN in conjunction with the distances to two predetermined points, one in and one out of the xz-plane, for each remaining SN. This will be a focus of future work.

As discussed in Refs. [23,24], descriptors based on distances between arbitrarily indexed ions as well as angles between pairs of ions will not be able to uniquely define the configuration of ions if multiple ions are equidistant to the target ion for which the prediction is being made. With the descriptors developed here, there is a similar inability to properly handle SN configurations if two or more SNs are equidistant to the FN as there is an ambiguity in the nearest SN ranking. In this case, the SN configuration lives on the boundary of the domain of all physically achievable SN descriptor vectors. During a MD simulation, as the configuration of SNs passes through an instance where there is an ambiguity in the nearest SN ranking, the SN descriptor vector will leave the domain of possible SN descriptor vectors and simultaneously re-enter the domain at a new location. Over the course of this sudden change in the nearest SN ranking, both the ion positions, in Cartesian coordinates, and the underlying total electron density around the FN will change continuously. This means the two points on the boundary of the domain of possible SN descriptors connected by the sudden change in the nearest SN ranking have the same target FN weight. Therefore, any perturbation of the equidistant SNs can be performed allowing for the configuration of SN with an ambiguity in the nearest SN ranking to be approximate by a new configuration of SNs that is uniquely defined. A numerical check of this will be presented in the results section.

III. Reference data

As proof of principle, the force-correction model will be trained and tested for warm dense hydrogen at various temperatures along the 1.0-g/cm^3 isochore. The range of temperatures for which the force correction model is applicable is bounded at both ends. The upper bound on this temperature range will be determined by the convergence of the free energies of OF-DFT and KS-DFT. Reference [43] showed that for deuterium this convergence will have occurred by

200 kK. Above this temperature, the upfront cost of the force correction model is not worth the minimal gain in accuracy over OF-DFT. In the case of the lower-temperature bound, it is expected that the base assumption of a correction being easier to learn will begin to break down. An estimate of this temperature will be determined by constructing models at various temperatures between 10 and 150 kK.

For each of the temperatures considered, a single reference data set will be constructed such that the ionic configurations and corresponding forces are consistent to that particular temperature. No mixing of ionic configurations and forces generated at other temperatures will be allowed. To generate the reference ionic configurations for a given temperature, OF-DFT based MD (OFMD) is performed. This is done under the assumption that due to the lack of ionic structure at the given temperatures, OFMD will sample the domain of local configurations consistent with KS-DFT–based MD (KSMD). This assumption will begin to break down at temperatures below 3 kK when molecular hydrogen begins to form [44] (this is the motivation for limiting the lowest temperature considered to 10 kK). Furthermore, by using OFMD to generate the local configurations the upfront cost of collecting reference data for the model will be significantly reduced.

All OFMD simulations are performed in a canonical ensemble using an Andersen thermostat with a time step of between 0.05 and 0.24 fs. To further reduce the upfront cost of the model, a system of 20 hydrogen atoms in a periodic cubic supercell is used. In total, each OFMD trajectory will consist of 18k steps. From an OFMD trajectory, a snapshot will be extracted every 30 steps to form a reference data set consisting of 600 snapshots. Once extracted, a single point calculation will be performed to generate the reference force differences for each snapshot. Of the 600 snapshots sampled the first 500 snapshots are taken as the master training set. The remaining 100 are set aside as the testing set and will be used only once all training processes are completed. It should be noted that all 20 local configurations will be used from each snapshot, giving the master training and test sets 10k (90 and 150 kK have 5k) and 2k local configurations, respectively.

All DFT calculations were performed with *PROFESS@Q-ESPRESSO* [5], a coupling of the OF-DFT based *PROFESS* package [45,46] with the KS-DFT based *QUANTUM ESPRESSO* package [47,48]. KS calculations use a 6 × 6 × 6 automatically generated Monkhorst–Pack k mesh with a plane wave energy cutoff between 600 to 800 Ry and 40 to 150 bands, depending on the system temperature. OF-DFT calculations are carried out with a 64 × 64 × 64 real space grid. The noninteracting free energy is treated with the LKTFγTF functional [49]. This is a convex combination of the finite temperature LKTF [11] and finite temperature Thomas–Fermi (TF) [50] noninteracting free-energy functionals. The convex parameter γ was arbitrarily chosen to be 0.5. Moreover, both OF and KS calculations employ the ground state SCAN-L [51,52] exchange correlation functional and utilize an equivalent local pseudopotential [53].

The force correction model has been written in *Python* using *NumPy* [54] and parallelized with *mpi4py* [55–58]. The *Python* code was then interfaced with *Quantum Espresso* [47,48]. At present, the interfacing of the force correction model requires the reference data collection, training process and use in MD to be done in three separate steps. Work is ongoing to implemental the model in a way that enables on-the-fly learning and predictions.

IV. Results

A. Training the model

Each force correction model is constructed using a single layer fully connected feed-forward NN with a the hidden layer consisting of 40 to 100 nodes. The training run for an individual model begins with a random initialization of the free parameters on the range [-0.1, 0.1]. The cost of Eq. (11) is then minimized using a gradient descent with a learning rate on the order of 10^{-3}. Details concerning the backpropagation used to minimize Eq. (11) can be found in the supplemental material [42]. Typical training runs require 50 k to 100 k epochs to optimize the NN. The hyperparameters A and α of the regularization term are set to 300 and 10^{8}, respectively for all training sets.

In order to produce a reliable model both the cutoff radius and the number of SNs in the construction of the descriptor vector must be optimized. The cutoff radius was chosen by examining how the predicted FN weights behave as a function of the FN distance to the reference ion. As can be seen in Fig. 2 (a) for the 90 kK data set with the use of 3 SNs, when the cutoff radius is set to 3.78 bohr the FNs near the cutoff radius will have a weight that is three orders of magnitude smaller than the weights of the FNs closest to the reference ion. In effect, new ions passing into the volume defined by the cutoff radius will cause a negligible jump in the predicted force difference. Moreover, the decay of the FNs weight shown in Fig. 2 (a) confirms the assumption that the force difference, in the case of hydrogen, only depends on the local configuration of neighboring ions.

To determine the number of SNs needed in the construction of the descriptor vectors, a series of convergence test of the model's accuracy were performed. When 1 or 2 SNs are used for the 90 kK data set, the average relative error in the predicted KS force magnitude is around 10%, Table I. When 3 SNs are used the average relative error in the predicted KS force magnitude drops to around 7%. Increasing the number of SNs to 6 SNs can reduce the average relative error in the predicted KS force magnitude to around 6%. This additional gain in accuracy going from 3 to 6 SNs primarily comes from improvement in the smallest predicted KS forces which experience the largest relative errors. The results of this convergence test were consistent for other temperatures considered. Moving forward, 3 SNs will be used to construct the force correction model at 60, 90 and 150 kK. In the case of 10 and 30 kK 5 SNs are used in an attempt to improve the predictions of the smallest KS forces.

Table I: Result from the convergence tests used to determine the number of SNs needed in the construction of the descriptor vectors. Column 1 indicates the number of SNs used. Column 2 is the average relative error in the predicted KS force magnitude. Columns 3 through 5 provide the values of three different percentiles of the relative error distribution. Column 6 is the average angular deviation of the predicted KS force with respect to the target KS force. Columns 7 through 9 provides three different percentiles of the distribution of angular errors.

# SNs	Rel. Err. (%)	25th (%)	50th (%)	75th (%)	Ang. Err. (degrees)	25th (degrees)	50th (degrees)	75th (degrees)
1	9.25	5.21	8.26	12.10	2.81	1.05	2.02	3.45
2	10.75	5.64	9.03	13.38	3.72	1.33	2.51	4.60
3	7.20	3.54	5.86	9.10	2.74	1.10	1.94	3.38
4	7.35	4.11	5.84	8.48	2.56	0.98	1.80	3.16
5	6.27	3.08	4.71	7.26	2.62	1.00	1.82	3.11
6	6.00	2.93	4.41	7.27	2.43	0.95	1.70	2.95
7	6.90	3.76	5.45	8.09	2.51	1.00	1.77	3.19

With the cutoff radius and number of SNs set, the next step is to produce the ensemble of models needed to obtain an uncertainty measure for MD simulations. To train each individual force-correction model in the ensemble, a random sampling of the master training set is performed such that 58% of all of the data forms the training set, 25% forms the validation set, and the remaining 17% forms the pseudo-test set. The training set is used to determine the optimal free parameters. The validation set is used to monitor and terminate the training process in an attempt to avoid overfitting. The pseudo-test set is used at the end of the training process to obtain an estimate of the generalization error of the individual model in the ensemble. Once all 15 force-correction models of the ensemble have been trained, the test set described in Sec. III is used to determine the generalization error of the ensemble.

At 90 kK, the predictions with the ensemble of models are in good agreement with the target KS values and are a significant improvement over the predictions of OF-DFT, Fig. 3 row two column 1. In terms of the relative errors, 60% of the predicted KS force magnitudes are within 5.2% of the target KS value (a level of error that is consistent with previous works [26,59,60]). Comparatively, only 5% of the pre-corrected OF forces in the test set have magnitudes within 14% of their respective target KS values. When the relative errors are examined further, the smallest relative errors of the model occur for the largest target force magnitudes. As the target force magnitude decreases, the accuracy of the model worsens, leading to the smallest target forces being predicted with a relative error of 20% or more. In terms of the angular error, 60% of the pre-corrected OF forces in the test set deviate from their corresponding reference KS force by less than 4.4°. While the OF force provides a good starting direction for the predicted KS force, the force correction model is able to further reduce the angular error to 2° error or less for 60% for the predicted KS forces. For further discussion of other temperatures see the supplemental material [42].

The last step before using the model in the context of MD is to confirm that an ambiguity in the SN rankings can be handled perturbatively so that a unique SN ranking can be produced. To test this, SN configurations from the test set at 90 kK were taken and the ion positions were modified so that the configuration of SNs just before and just after an ambiguity in the nearest SN ranking could be obtained. For all configurations considered the nearly equidistant SNs are offset by 0.002 bohr. The resulting predicted FN weights before and after the change in the SN ranking are shown in Fig. 2 (b).

As can be seen in Fig. 2 (b), the largest FN weights are nearly consistent with one another before and after the change in the SN ranking. More specifically, the difference in the largest FN weights before and after the ambiguity in the nearest SN ranking is around 1% of the FN weight just before the ambiguity. For the smaller FN weights the consistency between the FN weight before and after is significantly worse and in many cases the difference can exceed 100% of the before weight. This is likely the result of the FN weights being learned implicitly through the force differences causing the largest FN weights to dominate the cost. While the consistency of the FN weights before and after does not hold for all predicted weights, the impact of the inconsistency of the smallest FN weights during a perturbation to define a unique nearest SN ranking will be inconsequential for the total predicted force difference as the largest FN weights correlate well with FNs closest to the reference ion and the reference force differences are well correlated to the nearest FN, Fig. 2 (c).

B. Model comparisons

Before moving on to MD simulations, the newly developed force correction model is benchmarked against the Behler and Parrinello (BP) model [12]. For this benchmarking the BP model will be used to predict both force differences and KS forces directly. The BP model was directly implemented into our current code with all descriptor derivatives being taken analytically. For each of the following comparisons the same training and testing set at 90 kK describe above in section IV subsection A is used. Further details about the parameters of the BP models can be found in [42].

Comparison 1: The SN descriptors based on inter-ionic distances forming $\vec{d}^{T}_{\mathrm{SN}^{(j)},j}$ are swapped out for the BP descriptors (Eqs. (4) and (5) of [12]) within the force correction model developed here. After varying the number of descriptors and cutoff radius for the SN description, the optimal version of the model used 6 radial and 4 angular descriptors and had a 3.78 bohr SN cutoff radius. The resulting median relative error in the predicted KS force magnitudes is 6.82% and the median angular error of the predicted KS force is 2.34 degrees, table II row 2. With respect to the force correction model developed within this work (row one of table II), the use of the BP descriptors results in comparable error distributions.

Comparison 2: The full BP model of [12] is used to predict force differences. To maintain as fair of a comparison as possible, only force data is used to train the model (inclusion of energy data in the cost function diminished the accuracy of the final predicted KS forces [42]). After testing multiple sets of descriptors and cutoff radii [42] the optimal predicted KS forces were found to be given by a model with a 3.78 bohr cutoff radius that used 2 radial and 4 angular descriptors. The resulting predicted KS forces, table II row 3, has a median relative error in the magnitude of 9.5% and a median angular error of 2.9 degree. Both errors are slightly higher then the force correction model based on the developed framework.

Comparison 3: The full BP model of [12] is used to directly predict the KS forces. Similar to comparison 2, only force data is used in the training process. During the optimization of the hyperparameters it was found that the accuracy of the predicted KS forces did not change significantly for cutoff radii between 5.6 and 17 bohr and for 55 to 65 descriptors [42]. For the comparison a BP model with a cutoff radius of 9.45 bohr with 30 radial and 24 angular descriptors was used. Compared to the previous 3 models based on force differences, the accuracy in the directly predicted KS forces is significantly worse with errors in the KS force magnitude of 70% or more and angular errors in excess of 70 degrees, table II row 4.

Overall, comparisons 1 and 2 demonstrate that within the framework of force differences the model developed in this work is capable of delivering the same level of accuracy as existing schemes. In comparison 3, the large errors are likely a result of insufficient data as the number of local configurations in the training set is significantly smaller than the typical hundreds of thousands or even millions of local configurations needed to train a model for the direct prediction of KS forces. This result highlights the benefit of the delta-learning approach as less data is needed to achieve reasonable errors.

Comparing the BP model to the model developed in this work further, there are other clear tradeoffs. First, the BP scheme requires a significant number of hyperparameters in the construction of the descriptors. The number of which increases as the number of descriptors increases. In the model developed here, the number of hyperparameters is always 2. This has the potential of speeding up the training process by reducing the size of the domain of hyperparameters that needs to be searched for the optimal model. Second, the BP model has the advantage that the forces are defined as the gradient of a scalar quantity meaning the resulting force field is conservative. Examination of the curl of the KS forces predicted by the mode developed here indicates that the resulting force field is in fact not conservative [42]. However, examining the change in energy during a MD simulation [42] indicates the non-conservative nature of the resulting force-field will be inconsequential when a thermostat is used. Nevertheless, the impact of the non-conservative nature of the forces needs to be investigated to ensure the a reasonable distribution of energies is predicted during MD simulations resulting in both an accurate mean energy and an accurate standard deviation needed for the calculation of specific heat.

Table II: Resulting errors in the predicted KS forces at 90 kK for the comparison with various Behler and Parinello [12] models. Column 1 indicates the direct output of each model. Column 2 is the average relative error in the predicted KS force magnitude. Columns 3 through 5 provide the values of three different percentiles of the relative error distribution. Column 6 is the average angular deviation of the predicted KS force with respect to the target KS force. Columns 7 through 9 provide three different percentiles of the distribution of angular deviations.

Model	Rel. Err. (%)	25th (%)	50th (%)	75th (%)	Ang. Err. (degrees)	25th (degrees)	50th (degrees)	75th (degrees)
$\Delta\vec{F}$ (this work)	7.20	3.54	5.86	9.10	2.74	1.10	1.94	3.38
$\Delta\vec{F}$ (BP des.)	8.19	4.80	6.82	9.92	3.08	1.26	2.34	3.83
$\Delta\vec{F}$ (full BP)	12.23	5.75	9.52	15.88	4.07	1.48	2.90	5.04
$\vec{F}^{KS}$(full BP)	154.52	82.52	120.73	187.98	75.46	42.78	70.85	110.08

C. Molecular dynamics

For the force-correction model to be a useful tool in the context of MD simulations, both the energy and pressure must be obtainable. In the case of pressure, the standard approach is to use the virial expression which utilizes the forces and positions of the ions [61]. However, Ref. [61] showed in the case of periodic systems an additional correction associated with the change in energy with respect to the change in simulation cell side length must be added to the standard virial expression. This additional correction term was later shown to involve partial forces associated with interactions between an ion and image ions [62]. Since the force-correction model provides only the total force on an ion, an alternative approach to obtaining the pressures is needed. [Note: initial attempts to use the individual terms of Eq. (9) within the context of [61] were unsuccessful; this is still an ongoing area of work where the model can be improved.]

To determine equivalent KS pressures from the model the corresponding OF pressures will be used. This can be done as shown in Fig. 4. Here, the reference KS pressures are plotted against the corresponding OF pressure for each snapshot that comprises the full reference data set of the 90-kK system. As can be seen there is a strong linear correlation between the KS and OF pressures. This is also true of the corresponding energies [Fig. 4 (b)]. Note, neither the energy nor pressure contains the ideal gas contribution from the ions. These correlations between OF and KS quantities weaken as the temperature of the system is decreased but exist even at 10 kK [Figs. 4 (c) and (d)]. As such, these linear correlations will be used to generate the corresponding distribution of KS energies and pressures when the force-correction model is used to drive MD simulations. Moreover, the spread in the reference data in the plots of Fig. 4 will be used to define an uncertainty in energy and pressure. This is done by shifting the best fit line symmetrically up and down until the region containing 95% of the reference data is found.

Shown in Fig. 3 are the resulting distributions of energies and pressure at each temperature considered for MD simulations driven by the force-correction model, KS-DFT and OF-DFT. Between 60 and 150 kK, the energy and pressure distributions from the force-correction model are in excellent agreement with the corresponding distributions from KS-DFT. As the temperature of the system decreases further to 30 and 10 kK, the distributions from the force-correction model begin to shift away from the target KS distribution and develop longer tails. However, overall the distributions of the model's predicted pressure and energies are still in good agreement to those of KS-DFT below 60 kK and are significant improvements over the distribution obtained with OF-DFT.

To further quantify the resulting energies and pressures from the MD simulations, the average of each distribution is compared to the corresponding average value obtained from the

KS distribution (Fig. 5). In the case of the energies [Fig. 5 (a)], the average from the force-correction model is consistently within 1% of the target KS energy for all temperatures. Even when the uncertainty associated with the fitting procedure of Fig. 4 is accounted for, the relative error in the energies from the model do not exceed 1.2%. This is an improvement over OF-DFT, which has relative errors above 3% for temperatures below 90 kK. For the pressures [Fig. 5 (b)], the relative errors from the force-correction model are higher than for the energies, but the relative error in pressure is typically within 2% of the target KS values. Again this is an improvement over OF-DFT, which has relative errors up to 7% at 10 kK.

Table III: Computational time needed to perform one MD step with KS and OF DFT. Note the cost of OF-DFT at 30 and 90 kK are the same. Also provided is the cost per training cycle for the NN.

Calculation	Time (s)
KS-DFT (at 90 kK)	632.32
KS-DFT (at 30 kK)	263.10
OF-DFT	7.89
Training NN (per epoch of training)	1.34

Finally, the total computational cost of MD simulations performed with the force-correction model is determined and compared to the cost of standard KS and OF DFT based MD. To provide a fair comparison, the computational cost of each method is estimated using 64 cpus. The resulting cost per MD step for OF and KS DFT is shown in Table III. For the calculation of the total cost of the force correction model the upfront cost associated with training the ensemble of models needs to be estimated. This upfront cost will be strongly dependent on the size of the training set needed to maintain a similar level of accuracy on each member of the ensemble. To optimize the training set size a series of convergence tests were performed using a single force correction model. The resulting errors, Table IV, at 90 kK indicate that the training set size can be reduced from current size of 4500 reference ions down to 1800 reference ions without significant loss in model accuracy. A similar test for 30 kK also indicated the current training set could be reduced by a factor of 2.

Table IV: Errors in the predict KS force at 90 kK as a function of the training set size. Column 1 indicates the number of local configurations in the training set. Column 2 is the average relative error in the predicted KS force magnitude. Columns 3 through 5 provide the values of three difference percentiles of the relative error distribution. Column 6 is the average angular deviation the predicted KS force has with respect to the target KS force. Columns 7 through 9 provide three different percentiles of the angular error distribution.

Training set size	Rel. Err. (%)	25th (%)	50th (%)	75th (%)	Ang. Err. (degrees)	25th (degrees)	50th (degrees)	75th (degrees)
125	19.41	7.22	13.17	23.01	8.01	2.71	5.50	10.01
375	10.43	4.72	8.55	13.04	4.23	1.66	2.88	5.22
625	10.23	5.33	9.37	13.23	3.48	1.35	2.44	4.21
1875	8.35	4.53	6.81	10.03	2.83	1.04	1.99	3.51
4500	7.20	3.54	5.86	9.10	2.74	1.10	1.94	3.38

When the total cost of performing KS single point calculations, generating the reference ion configuration with OFMD and the training of all 15 members of the ensemble are accounted for, the total up front cost for the force correction models at 30 and 90 kK are 1.16 and 1.18 million seconds respectively. The breakeven points associated with the upfront cost are equivalent to 4600 KSMD steps at 30 kK and 1900 KSMD steps at 90 kK. If the number of members in the ensemble is cut from 15 to 7 the breakeven points shift to 2200 steps and 1100 KS MD steps for 30 and 90 kK respectively. While the total upfront cost of the force correction model is still a significant portion of the cost of feasible KSMDs, once this upfront cost is paid the remaining cost will be equal to that of OFMD, making the force correction model a useful tool for simulations of WDM systems.

V. Summary

A ML based model has been constructed to correct OF-DFT calculated ionic forces to produce corresponding KS ionic forces. This was done by first constructing an approximate force difference in terms of the ionic positions. Here the ionic positions were used in two ways, First, as a set of grid points used to resolve the underlying electron density difference. Second, the distances between all ions of the SN configuration were used to form the input vector to a NN. It was discussed that by using all distances in the SN configuration in conjunction with an indexing scheme based on the nearest SN ranking, the SN configuration could be uniquely determined up to a rotation and reflection when no ambiguity in the ranking exist. In the case of an ambiguity in the nearest SN ranking, tests indicated that a perturbative approach can be taken to provide an approximate SN configuration that can be uniquely defined.

The resulting model was trained and applied to warm dense hydrogen between 10 and 150 kK at 1.0 g/cm^3. An analysis of the errors demonstrated the KS force magnitudes can be learned within 5% and the direction can be learned within 5°. Once trained and tested, MD simulations were performed at various temperatures for warm dense hydrogen with the force correction model. The resulting energies and pressure are consistently within 1% and 2%, respectively, of their target KS values down to 10 kK. Finally, once the number of training samples were optimized the computational cost of the model was estimated, suggesting the current breakeven point with KSMD is around 1100 steps at 90 kK and 2200 steps at 30 kK.

As the main results of this work are a proof of principle, the goal of future works will be to explore avenues for further improvements. This will include using the uncertainties as an on-off switch for the model allowing for a controlled extrapolation to bring a system back within the domain of the descriptor vector space associated with the training set as well as extending the descriptor to multicomponent systems. Furthermore, while it was not shown here, the off diagonal components of the stress tensor show similar correlations between the KS and OF calculations. As such, it is expect that the force correction model will be a useful tool for the calculation of viscosities which are at present time not possible to obtain with standard ab-initio MD simulations at the temperatures considered.

All reference data used to train the models in this work along with the final parameters of the trained neural networks have been made publicly available at, https://github.com/jhinz2/ML_force_correction_model_development.git. All of the code developed in this work is available upon request with the corresponding author.

Acknowledgement

All authors were supported by the Department of Energy National Nuclear Security Administration under Award Number DE-NA0003856, the University of Rochester, the New York State Energy Research and Development Authority, and U.S. National Science Foundation PHY Grant No. 2205521 . Partial funding for S. X. H. was provided by the NSF Physics Frontier Center Award PHY-2020249. All computations were performed on the Laboratory for Laser Energetics HPC systems.

The authors also thank Qi Yu, Dayou Yu, and Wei Shi Shi from the mining group at the Rochester Institute of Technology for their useful discussions on machine learning.

Disclaimer

This report was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.

References:

1. P. Hohenberg and W. Kohn, Inhomogeneous electron gas, Phys. Rev. **136**, B864 (1964).

2. V. G. Kohn, On the theory of reflectivity by an x-ray multilayer mirror, Phys. Status Solidi B **187**, 61 (1995).

3. K. Burke and L. O. Wagner, DFT in a nutshell, Int. J. Quantum Chem. **113**, **96** (2013); 1601(E) (2013)

4. N. D. Mermin, Thermal properties of the inhomogeneous electron gas, Phys. Rev. **137**, A1441 (1965).

5. V. V. Karasiev, T. Sjostrom, and S. B. Trickey, Finite-temperature orbital-free DFT molecular dynamics: Coupling PROFESS and QUANTUM ESPRESSO, Comput. Phys. Commun. **185**, 3240 (2014).

6. V. V. Karasiev, T. Sjostrom, D. Chakraborty, J. W. Dufty, K. Runge, F. E. Harris, and S. B. Trickey, Innovations in finite-temperature density functionals, in *Frontiers and challenges in warm dense matter*, edited by F. Graziani, M. Desjarlais, R. Redmer, and S. Trickey, Lecture notes in computational science and engineering, vol. 96, edited by T. J. Barth, M. Griebel, D. E. Keyes, R. M. Nieminen, D. Roose, and T. Schlick (Springer, Heidelberg, 2014), p. 61.

7. R. P. Feynman, N. Metropolis, and E. Teller, Equations of State of Elements Based on the Generalized Fermi-Thomas Theory, Phys. Rev. 75, 1561 (1949).

8. F. Perrot, Gradient Correction to the Statistical Electronic Free-Energy at Non-Zero Temperature- Application to Equation-of-State Calaculations, Phys. Rev A 20 (2), 586 (1979).

9. V. V. Karasiev, T. Sjostrom, and S. B. Trickey, Generalized-gradient-approximation noninteracting free-energy functionals for orbal-free density functional calculations, Phys. Rev. B 86, 115101 (2012).

10. V. V. Karasiev, D. Chakraborty, O. A. Shukruto, and S. B. Trickey, Nonemperical generalized gradient approximation free-energy functional for orbital-free simulations, Phys. Rev. B 88, 161108(R) (2013).

11. K. Luo, V. V. Karasiev, and S. B. Trickey, Towards accurate orbital-free simulations: A generalized gradient approximation for the noninteracting free energy density functional, Phys. Rev. B 101, 075116 (2020).

12. J. Behler and M. Parrinello, Generalized neural-network representation of high-dimensional potential-energy surfaces, Phys. Rev. Lett. **98**, 146401 (2007).

13. Z. Li, J. R. Kermode, and A. De Vita, Molecular dynamics with on-the-fly machine learning of quantum-mechanical forces, Phys. Rev. Lett. **114**, 096405 (2015).

14. V. Botu, R. Batra, J. Chapman, and R. Ramprasad, Machine learning force fields: Construction, validation, and outlook, J. Phys. Chem. C **121**, 511 (2017).

15. A. P. Bartók, R. Kondor, and G. Csányi, On representing chemical environments, Phys. Rev. B **87**, 184115 (2013).

16. V. L. Deringer, M. A. Caro, and G. Csányi, Machine learning interatomic potentials as emerging tools for materials science, Adv. Mater. **31**, 1902765 (2019).

17. L. Fiedler, K. Shah, M. Bussmann, and A. Cangi, Deep dive into machine learning density functional theory for materials science and chemistry, Phys. Rev. Mater. **6**, 040301 (2022).

18. P. Friederich, F. Häse, J. Proppe, and A. Aspuru-Guzik, Machine-learned potentials for next-generation matter simulations, Nat. Mater. **20**, 750 (2021).

19. J. Behler, Representing potential energy surfaces by high-dimensional neural network potentials, J. Phys.: Condens. Matter **26**, 183001 (2014).

20. G. Imbalzano, A. Anelli, D. Giofré, S. Klees, J. Behler, and M. Ceriotti, Automatic selection of atomic fingerprints and reference configurations for machine-learning potentials, J. Chem. Phys. **148**, 241730 (2018).

21. R. Jinnouchi, F. Karsai, C. Verdi, R. Asahi, and G. Kresse, Descriptors representing two- and three-body atomic distributions and their effects on the accuracy of machine-learned inter-atomic potentials, J. Chem. Phys. **152**, 234102 (2020).

22. N. Artrith, A. Urban, and G. Ceder, Efficient and accurate machine-learning interpolation of atomic energies in compositions with many species, Phys. Rev. B **96**, 014112 (2017).

23. F. Musil, A. Grisafi, A. P. Bartok, C. Ortner, G. Csanyi, and M. Ceriotti, Physics-inspired Structural Representations for Molecules and Materials, Chem. Rev. 121, 9759−9815 (2021).

24. S. N. Pozdnyakov, M. J. Willatt, A. P. Bartok, C. Ortner, G. Csanyi, and M. Ceriotti, Incompleteness of Atomic Structure Representations, Phys. Rev. Lett. 125, 166001 (2020).

25. T. D. Huan, R. Batra, J. Chapman, S. Krishnan, L. Chen, and R. Ramprasad, A universal strategy for the creation of machine learning-based atomistic force fields, npj Comput. Mater. **3**, 37 (2017).

26. N. Kuritz, G. Gordon, and A. Natan, Size and temperature transferability of direct and local deep neural networks for atomic forces, Phys. Rev. B **98**, 094109 (2018).

27. A. P. Bartók, J. R. Kermode, N. Bernstein, and G. Csányi, Machine learning a general-purpose interatomic potential for silicon, Phys. Rev. X **8**, 041048 (2018).

28. R. Jinnouchi, F. Karsai, and G. Kresse, On-the-fly machine learning force field generation: Application to melting points, Phys. Rev. B **100**, 014105 (2019).

29. S. K. Natarajan and J. Behler, Neural network molecular dynamics simulations of solid–liquid interfaces: Water at low-index copper surfaces, Phys. Chem. Chem. Phys. **18**, 28704 (2016).

30. T. E. Gartner, L. Zhang, P. M. Piaggi, R. Car, A. Z. Panagiotopoulos, and P. G. Debenedetti, Signatures of a liquid–liquid transition in an ab initio deep neural network model for water, Proc. Nat. Acad. Sci. **117**, 26040 (2020).

31. B. Cheng, G. Mazzola, C. J. Pickard, and M. Ceriotti, Evidence for supercritical behaviour of high-pressure liquid hydrogen, Nature **585**, 217 (2020).

32. R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld, Big data meets quantum chemistry approximations: The Δ-machine learning approach, J. Chem. Theory Comput. **11**, 2087 (2015).

33. P. Liu, C. Verdi, F. Karsai, and G. Kresse, Phase transitions of zirconia: Machine-learned force fields beyond density functional theory, Phys. Rev. B **105**, L060102 (2022).

34. A. P. Bartók, M. J. Gillan, F. R. Manby, and G. Csányi, Machine-learning approach for one- and two-body corrections to density functional theory: Applications to molecular and condensed water, Phys. Rev. B **88**, 054104 (2013).

35. P. Pattnaik, S. Raghunathan, T. Kalluri, P. Bhimalapuram, C. V. Jawahar, and U. D. Priyakumar, Machine learning for accurate force calculations in molecular dynamics simulations, J. Phys. Chem. A **124**, 6954 (2020).

36. M. Born and R. Oppenheimer, Zur quantentheorie der molekeln, Ann. Phys. **389**, 457 (1927).

37. R. P. Feynman, Forces in molecules, Phys. Rev. **56**, 340 (1939): H. Hellmann, *Einfuhrung in die quantenchemie* (F. Deuticke, Leipzig und Wien, 1937).

38. Karasiev, V.V., Trickey, S.B. & Harris, F.E. Born–Oppenheimer Interatomic Forces from Simple, Local Kinetic Energy Density Functionals. J Computer-Aided Mater Des 13, 111–129 (2006).

39. V. Nair and G. E. Hinton, Rectified linear units improve restricted Boltzmann machines, in *Proceedings of the 27th International Conference on Machine Learning* (Omnipress, Haifa, Israel, 2010), p. 807.

40. A. A. Peterson, R. Christensen, and A. Khorshidi, Addressing uncertainty in atomistic machine learning, Phys. Chem. Chem. Phys. 19, 10978 (2017).

41. Y. Nagai, M. Okumura, Keita Kobayashi, and M. Shiga, Self-learning hybrid Monte Carlo: A first-principles approach, Phys. Rev. B 102, 041124 (2020).

42. See Supplemental Material at [insert link] for more information about the uniqueness of the descriptors, computational details for the calculation of the reference data, training of the model, additional predictions at various temperatures, model uncertainties during molecular dynamic simulations, further comparison with the Behler and Parrinello model and for further discussion on the non-conservative nature of the developed force field.

43. D. I. Mihaylov, V. V. Karasiev, S. X. Hu, J. R. Rygg, V. N. Goncharov, and G. W. Collins, Improved first-principles equation-of-state table of deuterium for high-energy-density applications, Phys. Rev. B 104, 144104 (2021).

44. J. Hinz, V. V. Karasiev, S. X. Hu, M. Zaghoo, D. Mejia-Rodrigues, S.B. Trickey, and L. Calderin, Fully consistent density functional theory determination of the insulator-metal transition boundary in warm dense hydrogen, Phys. Rev. Research 2, 032065(R) (2020).

45. G. S. Ho, V. L. Lignères, and E. A. Carter, Introducing PROFESS: A new program for orbital free density functional theory calculations, Comput. Phys. Commun. 179, 839 (2008).

46. M. Chen, J. Xia, C. Huang, J. M. Dieterich, L. Hung, I. Shin, and E. A. Carter, Introducing PROFESS 3.0: An advanced program for orbital-free density functional theory molecular dynamics simulations, Comput. Phys. Commun. 190, 228 (2015).

47. P. Giannozzi, S. Baroni, B. Nicola, N. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, G. L. Chiarotti, M. Cococcioni, I. Dabo et al., QUANTUM ESPRESSO: A modular and open source software project for quantum simulations of materials, J. Phys.: Condens. Matter 21, 395502 (2009).

48. P. Giannozzi, O. Andreussi, T. Brumme, O. Bunau, M. Buongiorno Nardelli, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, M. Cococcioni et al., Advanced capabilities for materials modelling with Quantum ESPRESSO, J. Phys.: Condens. Matter 29, 465901 (2017).

49. V. V. Karasiev, J. Hinz, and D. I. Mihaylov, "Tunable noninteracting free-energy functionals," unpublished.

50. R. P. Feynman, N. Metropolis, and E. Teller, Equations of state of elements based on the generalized Fermi-Thomas theory, Phys. Rev. 75, 1561 (1949).

51. D. Mejia-Rodriguez and S. B. Trickey, Deorbitalization strategies for meta-generalized-gradient-approximation exchange-correlation functionals, Phys. Rev. A **96**, 052512 (2017).

52. D. Mejia-Rodriguez and S. B. Trickey, Deorbitalized meta-GGA exchange-correlation functionals in solids, Phys. Rev. B **98**, 115161 (2018).

53. V. V. Karasiev, T. Sjostrom, and S. B. Trickey, Generalized-gradient approximation noninteracting free-energy functionals for orbital-free density functional calculations, Phys. Rev. B **86**, 115101 (2012).

54. C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith *et al.*, Array programming with NumPy, Nature **585**, 357 (2020).

55. L. Dalcín, R. Paz, and M. Storti, MPI for Python, J. Parallel Distrib. Comput. **65**, 1108 (2005).

56. L. Dalcín, R. Paz, M. Storti, and J. D'Elía, MPI for Python: Performance improvements and MPI-2 extensions, J. Parallel Distrib. Comput. **68**, 655 (2008).

57. L. D. Dalcin, R. R. Paz, P. A. Kler, and A. Cosimo, Parallel distributed computing using Python, Adv. Water Resour. **34**, 1124 (2011).

58. L. Dalcin and Y. L. L. Fang, Mpi4py: Status update after 12 years of development, Comput. Sci. Eng. **23**, 47 (2021).

59. T. Suzuki, R. Tamura, and T. Miyazaki, Machine learning for atomic forces in a crystalline solid: Transferability to various temperatures, Int. J. Quantum Chem. **117**, 33 (2017).

60. R. Tamura, J. Lin, and T. Miyazaki, Machine learning forces trained by Gaussian process in liquid states: Transferability to temperature and pressure, J. Phys. Soc. Japan **88**, 044601 (2019).

61. M. J. Louwerse and E. J. Baerends, Calculation of pressure in case of periodic boundary conditions, Chem. Phys. Lett. 421, 138 (2006).

62. A. P. Thompson, S. J. Plimpton, and W. Mattson, General formulation of pressure and stress tensor for arbitrary many-body interaction potentials under periodic boundary conditions, J. Chem. Phys. 131, 154107 (2009).

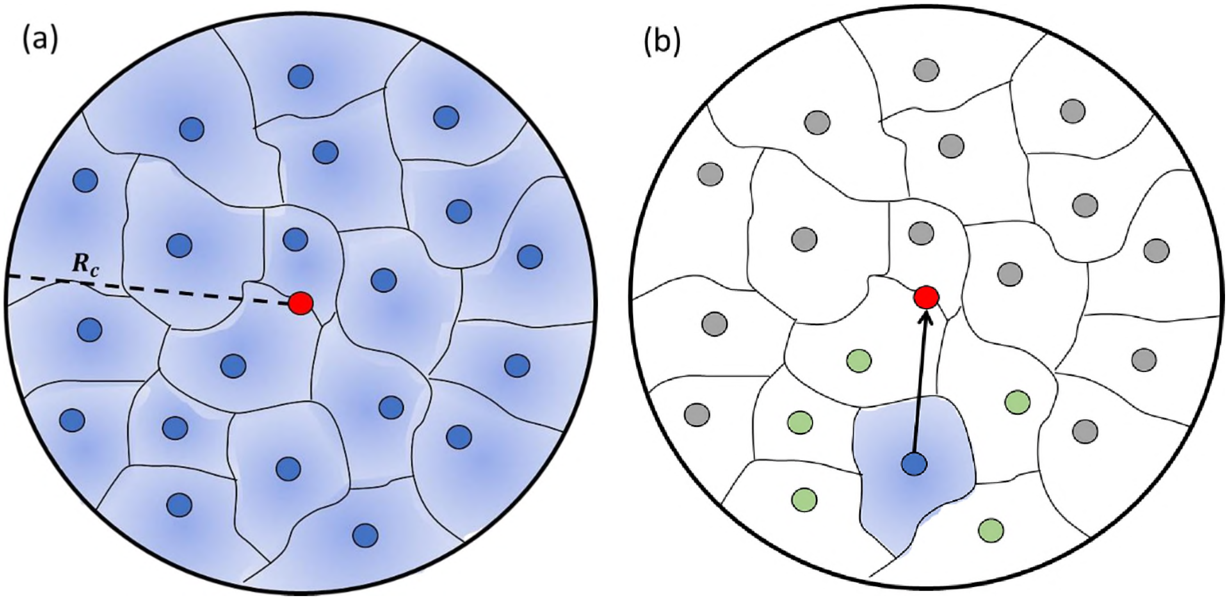


Fig. 1. (a), left: the neighboring ions shown as blue circles are used to divide the volume around the red reference ion into sub-volumes. Each sub-volume is constructed such that it contains exactly one neighboring ion. When a ion is used to indicate a sub-volume it will be referred to as a first neighbor (FN). The choice of the sub-volume boundaries, thin block line, was chosen arbitrarily for this example. (b), right: during the construction of the force correction model is assumed that the electronic density difference within a given sub-volume can be determined by the neighboring ions. Therefore, the contribution to the force difference of the ion in red from the shaded blue sub-volume, black arrow, will be determined by describing the configuration of green ions (referred to as second neighbors) in conjunction with the corresponding FN, blue circle. All other ions within the cutoff radius, gray circles, do not contribute to the weight of the sub-volume being considered.

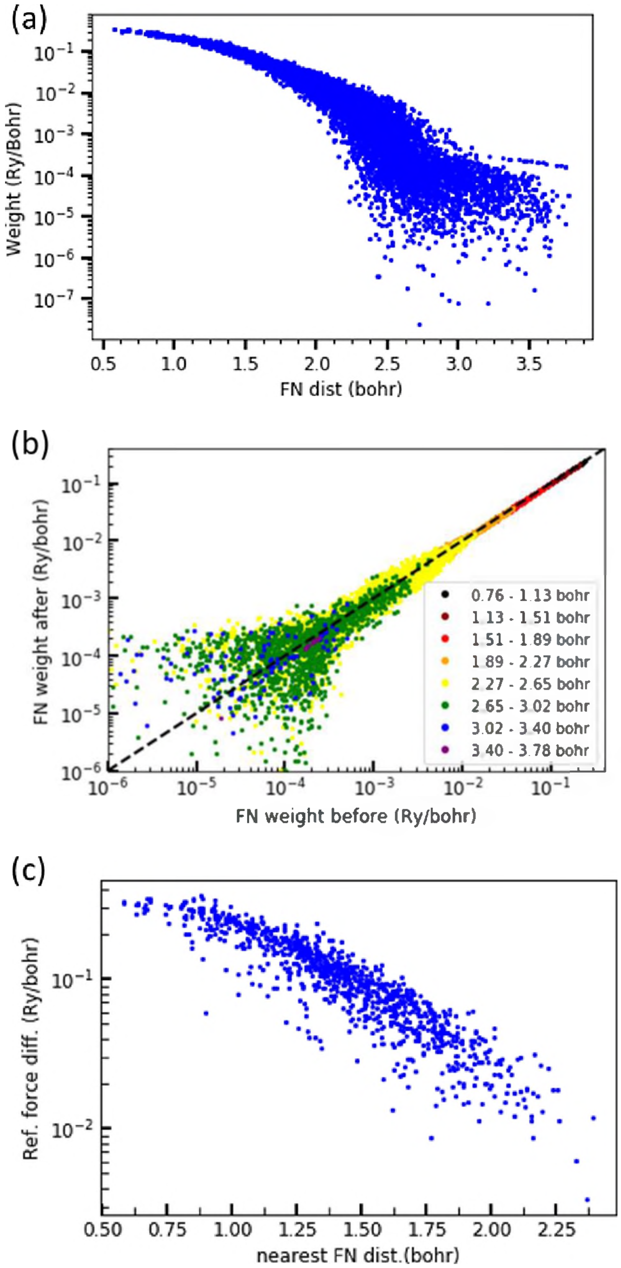


Fig. 2. (a), predicted FN weights as a function of the distance between the FN and reference ion for the 90 kK training. These predictions were performed with a force correction model that used 3 SNs and a cutoff radius of 3.78 bohr. (b), the predicted FN weights after a sudden change in the nearest SN ranking as a function of the predicted weights just before a sudden change in the SN ranking. The colors of the points indicate the distance of the FN to the reference ion. The black

dotted line represents the target line if the FN weights are continuous at an ambiguity in the nearest SN ranking. All predictions are also with a model using 3 SNs and a cutoff radius of 3.78 bohr. (c), reference force difference as a function of the nearest FN.

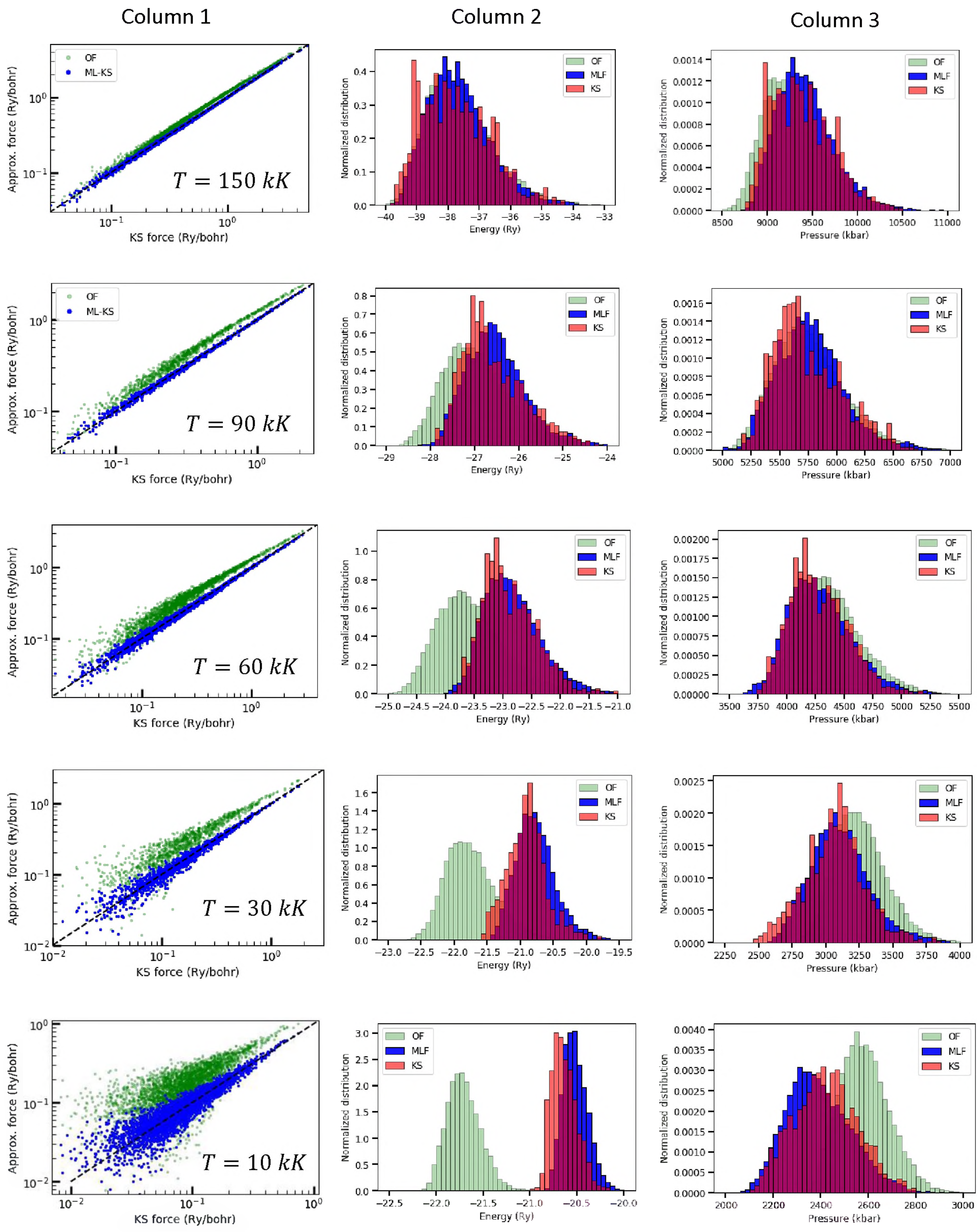

Column 1
Column 2
Column 3
OF
ML-KS
OF
MLF
KS
$T = 150\ kK$
$T = 90\ kK$
$T = 60\ kK$
$T = 30\ kK$
$T = 10\ kK$
Approx. force (Ry/bohr)
KS force (Ry/bohr)
Normalized distribution
Energy (Ry)
Pressure (kbar)

Fig. 3. Column 1 is the magnitude of an approximate force versus the magnitude of the target KS forces on the test set at each temperature. The points indicate the values of the underlying OF forces and the blue points are the predicted KS forces. The black line is again the target that the force-correction model aims to achieve. Column 2 is the distribution of energies (without ideal gas contribution from ions) for molecular dynamics performed with KS (red) and OF (green) DFT and the force-correction model (blue). Column 3 is the corresponding distributions for the pressures from MD simulations with the three methods (again with no contribution from ions). Each row corresponds to a different temperature. The first row begins at 150 kK and each subsequent row descends in temperature starting from 90 to 60 to 30 and to 10 kK, respectively. Note, all distributions in a given plot are binned on the same range with the same number of bins.

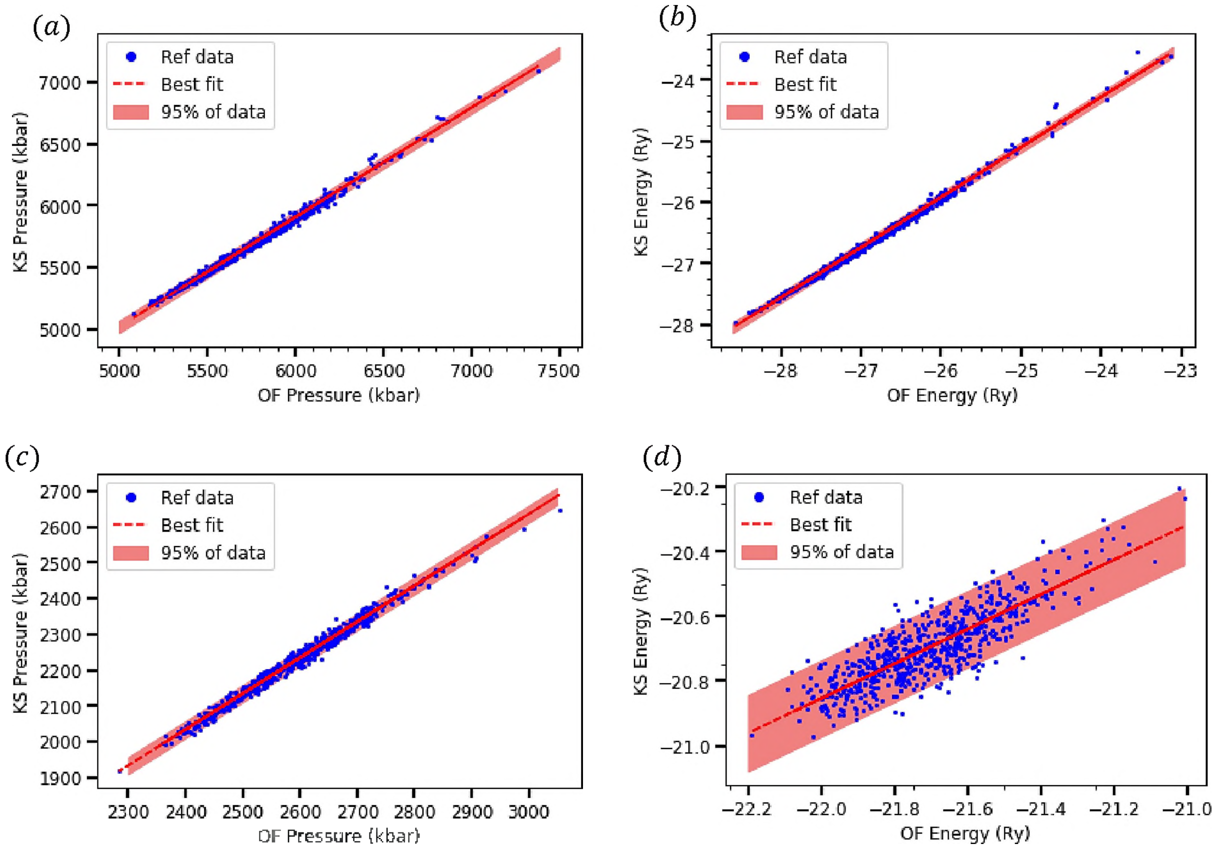


Fig. 4. (a) KS pressure and (b) KS energy plotted as functions of the corresponding OF quantity for each snapshot in the reference data set at 90 kK. (c) KS pressure and (d) KS energy plotted as functions of the corresponding OF quantity for each snapshot in the reference data set at 10 kK. The red dashed line in each plot is the best fit line of the reference data marked by the blue circles. The light red region indicates the area where 95% of the reference data falls around the best fit line.

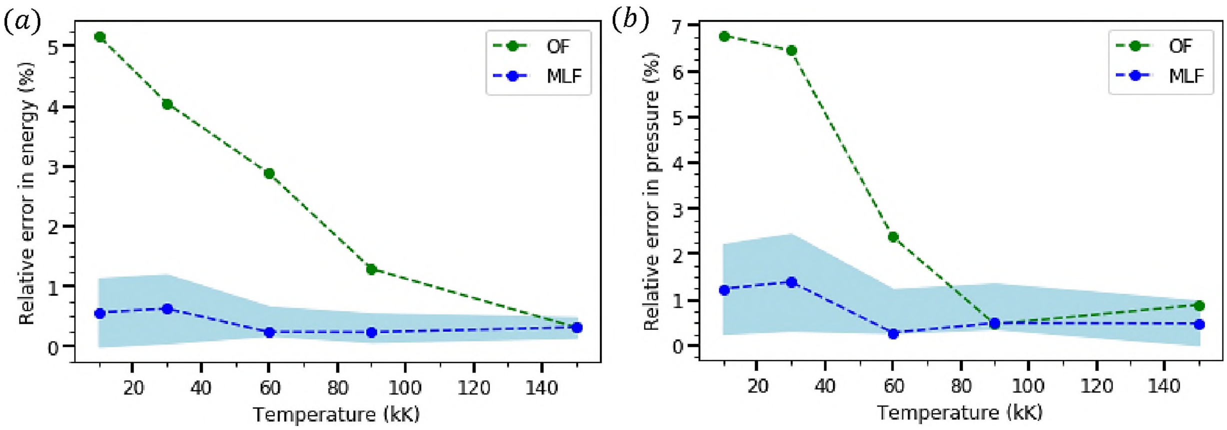


Fig. 5. (a) Relative error in the average pressure obtained from MD simulations. The blue circles are those from MD driven with the force-correction model and the green circles are from MD driven with OF-DFT. The dotted line connecting the points is a guide to the eye. (b) Corresponding relative errors in the average pressures. Note, all relative errors are calculated with respect to the KS average quantities. Furthermore, the blue region in each plot shows the effect the uncertainty from the fitting procedure shown in Fig. 4 has on the average energies and pressure in the case of the force-correction model.

Supplemental material: Development of a machine-learning–based ionic-force correction model for quantum molecular dynamic simulations of warm dense matter

Joshua P. Hinz[1], Valentin V. Karasiev[1], Suxing X. Hu[1,2], Deyan I. Mihaylov[1]

[1]Laboratory for Laser Energetics, University of Rochester, Rochester, NY 14623

[2]Department of Mechanical Engineering, University of Rochester, Rochester, NY 14623

I. Outline

The supplemental material is ordered as follows. Section II discusses the uniqueness of the descriptor vectors constructed in this work. In section III details about the parameters of the Kohn-Sham (KS) and orbital-free (OF) density functional theory (DFT) calculations can be found. Section IV provides a brief overview of the implementation of the force correction model into the existing software packages used for KS and OF calculations. In section V the details for the gradient descent used to optimize the neural network (NN) are discussed. This is then followed by the technical details and results of the training process in section VI. Section VII provides the uncertainty distribution from the molecular dynamic (MD) trajectories calculated with the force correction model. Section VIII contains the parameters for the Behler and Parrinello models used in the benchmarking. And finally, section IX contains the test to access the non-conservative nature of the force field predicted by the force correction model.

II. Uniqueness of descriptor vectors

The uniqueness of the descriptors used to describe a configuration of second neighbors (SN) with a unique nearest neighbor ranking to the first neighbor (FN) can be seen as follows. To begin, the FN associated with the SN configuration will be set at the origin. Since the configuration of SNs only needs to be uniquely defined up to a rotation, the configuration of SNs can be rotated such that the 1st SN (nearest to the FN) is positioned on the z-axis.

When the distance from the FN to the kth SN, with k > 1, is given, the kth SN must lie on the surface of a sphere centered on the FN with a radius $|\vec{R}_{FN,SNk}|$. Similarly, when the distance between the 1st SN and kth SN is given the kth SN must lie on the surface of a sphere centered on the 1st SN with a radius $|\vec{R}_{SN1,SNk}|$. Together, the kth SN's position must simultaneously satisfy

$$x_k^2 + y_k^2 + z_k^2 = |\vec{R}_{FN,SNk}|^2 \tag{1}$$

and

$$x_k^2 + y_k^2 + \left(z_k - |\vec{R}_{FN,SN1}|\right)^2 = |\vec{R}_{SN1,SNk}|^2. \tag{2}$$

By eliminating the x and y component of the kth SN between Eqs. (1) and (2) the z position of the possible locations of the kth SN can be determined;

$$z_k = \frac{1}{2|\vec{R}_{FN,SN1}|}\left(|\vec{R}_{FN.SN1}|^2 + |\vec{R}_{FN,SNk}|^2 - |\vec{R}_{SN1,SNk}|^2\right). \tag{3}$$

Substituting the result of Eq. (3) back into Eq. (1) gives

$$x_k^2 + y_k^2 = R_{FN,SNk}^2 - \frac{1}{4|\vec{R}_{FN,SN1}|^2}\left(|\vec{R}_{FN.SN1}|^2 + |\vec{R}_{FN,SNk}|^2 - |\vec{R}_{SN1,SNk}|^2\right)^2 \tag{4}$$

which describes a circle of possible positions of the kth SN.

For the kth SN with k > 2, the set of possible positions can be reduced further by adding the distance between the 2nd SN and kth SN to the aforementioned distances. Here the SN configuration will be rotated again such that the 2nd SN lies in the xz-plane and has a positive x value. The z value of the 2nd SN will still be given by Eq. (3) and the resulting x value will be given by Eq. (4) with y_k set to zero. In the case where the 2nd SN lies on the z axis after the initial rotation aligning the 1st SN with the z-axis, this additional rotation cannot be performed. In this case the next SN which does not lie on the z-axis will take the place of the 2nd SN in the following discussion. The result of the discussion will still hold due to the fact that all distances between ions will ultimately be used in the descriptor vector.

When the distance $|\vec{R}_{SN2,SNk}|$ is added to the two previous distances, $|\vec{R}_{FN,SNk}|$ and $|\vec{R}_{SN1,SNk}|$, the position of the kth SN now must simultaneously satisfy Eq. (4) and

$$(x_k - x_2)^2 + y_k^2 + (z_k - z_2)^2 = \left|\vec{R}_{SN2,SNk}\right|^2. \tag{5}$$

Here, both z values have already been determined in terms of distances between ions. By eliminating y_k between Eqs. (4) and (5), the resulting x_k value is given as

$$x_k = \frac{1}{2x_2}\left((z_k - z_2)^2 + x_2^2 + \left|\vec{R}_{FN,SNk}\right|^2 - \left|\vec{R}_{SN2,SNk}\right|^2 - z_k^2\right). \tag{6}$$

Substituting this value back into Eq. (1) gives two possible y_k values;

$$y_k = \pm\sqrt{\left|\vec{R}_{FN,SNk}\right|^2 - x_k^2 - z_k^2}. \tag{7}$$

In effect, by using the distances $|\vec{R}_{FN,SNk}|$, $|\vec{R}_{SN1,SNk}|$ and $|\vec{R}_{SN2,SNk}|$ for all SNs with the distances $|\vec{R}_{FN,SN1}|$, $|\vec{R}_{FN,SN2}|$ and $|\vec{R}_{SN1,SN2}|$ each SN's possible position is reduced to two possible locations.

More generally, any point $\vec{R}$ not along the z-axis will be equidistant to at most two points on a circle about the z-axis. These two points will be reflections of one another about the plane with normal in the direction $\vec{R}x\hat{z}$. This reflection symmetry has important consequences when the distance between the k^th^ and m^th^ SN is used with the previous given distances.

As discussed, the set of distances $\{|\vec{R}_{FN,SNk}|, |\vec{R}_{SN1,SNk}|, |\vec{R}_{SN2,SNk}|\}$ and $\{|\vec{R}_{FN,SNm}|, |\vec{R}_{SN1,SNm}|, |\vec{R}_{SN2,SNm}|\}$ reduces the kth and mth SN location down to two points each. This is illustrated in Fig. 1 (a) with the solid filled red and blue points representing the possible locations of the kth and mth SN respectively. For the moment the kth and mth are both assumed to not live in the xz-plane. If the kth SN is located at the point shown with a thick black boarder in Fig. 1 (a), the set of distances $\{|\vec{R}_{FN,SNm}|, |\vec{R}_{SN1,SNm}|, |\vec{R}_{SNk,SNm}|\}$ will also pick out at most two possible points for the mth SN indicated by the gray circles. However, the blue points are reflections of one another about the plane with normal in the direction $\vec{R}_{SN2}x\hat{z}$ and the gray points are reflections of one another about the plane with normal in the direction $\vec{R}_{SNk}x\hat{z}$. As a set of points cannot be reflections of one another about two different planes in three dimensions, the gray and blue points cannot coincide perfectly. Therefore, when the kth SN is at the position indicated in Fig. 1 (a), the set of distances $\{|\vec{R}_{FN,SNk}|, |\vec{R}_{SN1,SNk}|, |\vec{R}_{SN2,SNk}|, |\vec{R}_{FN,SNm}|, |\vec{R}_{SN1,SNm}|, |\vec{R}_{SN2,SNm}|, |\vec{R}_{SNk,SNm}|\}$ picks out a unique position for the mth SN. This means the four possible configurations of the kth and mth SN are reduced to two uniquely determined configurations and their respective reflections about the xz-plane.

Continuing in the same manner as discussed above. If the distances between adjacent SNs in the nearest SN ranking are included with the distances $\{|\vec{R}_{FN,SNk}|, |\vec{R}_{SN1,SNk}|, |\vec{R}_{SN2,SNk}|\}$ for each of the k SNs 3 through n (where n is the total number of SNs) and the distances

$\{|\vec{R}_{FN,SN1}|,\ |\vec{R}_{FN,SN2}|, |\vec{R}_{SN1,SN2}|\ \}$ and none of the SNs 3 through n live in the xz-plane, the configuration of SNs would be uniquely described. This can be seen graphically in Fig. 1 (b). By moving any SN from one of its possible locations to the other the distances between the adjacent SN will change. Similarly, if multiple SNs are moved between their respective possible positions at least one distance must change. Therefore, as long as all SNs do not live in the xz-plane the configuration of SN will be uniquely described up to a reflection about the xz-plane.

However, there are situations such as the one shown in Fig. 1 (c) where multiple SNs live in the xz-plane. In this case, a SN not in the xz-plane that is adjacent to two SNs that do live in the xz-plane, can be moved between its two possible locations without a change in the distances between SNs. This means different SN configurations would have the same descriptor vector. To restore uniqueness the most straightforward way is to ensure all SNs which do not live in the xz-plane are connected to all other SNs that do not live in the xz-plane. This can be achieved by using the distances between all pairs of ions.

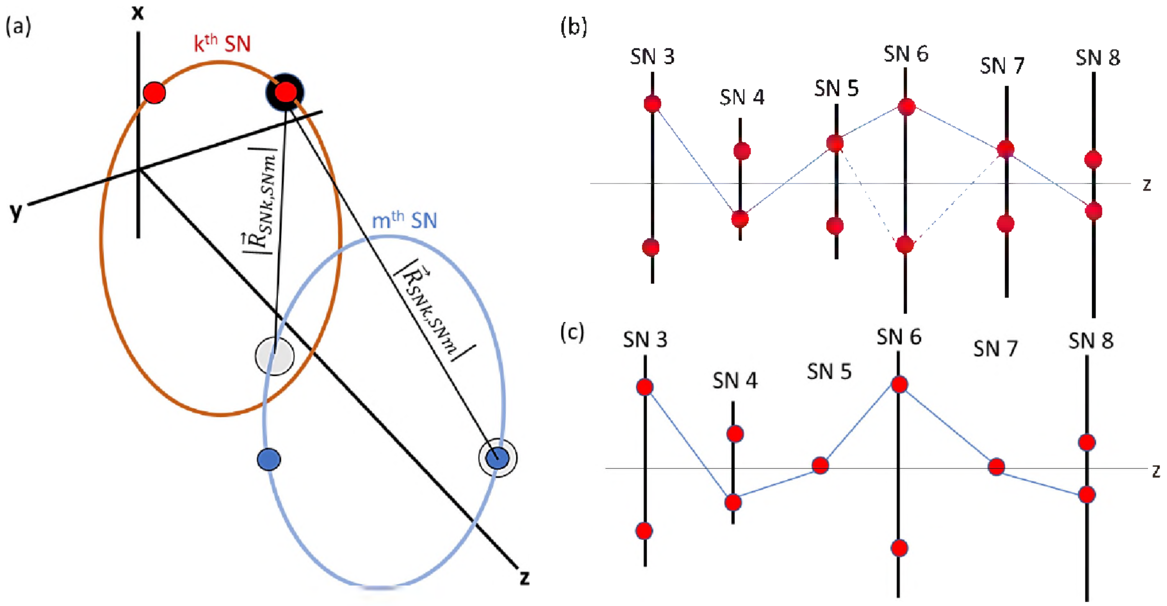


Figure 1: (a), The red filled circles indicate the two possible positions the kth SN can be found after the distances $\{|\vec{R}_{FN,SNk}|,\ |\vec{R}_{SN1,SNk}|, |\vec{R}_{SN2,SNk}|\ \}$ are given. Correspondingly, the blue filled points are the two possible locations for the mth SN resulting from the distance $\{|\vec{R}_{FN,SNm}|,\ |\vec{R}_{SN1,SNm}|, |\vec{R}_{SN2,SNm}|\ \}$. When the kth SN is found at the red circle with the thick black boarder the set of distances $\{|\vec{R}_{FN,SNkm}|,\ |\vec{R}_{SN1,SNm}|, |\vec{R}_{SNk,SNm}|\ \}$ picks out the gray filled circles as the two possible locations for the mth SN. (b), Graphical representation of connecting adjacent SNs in the nearest SNs ranking. The red points on the labeled black lines indicate the two possible positions of each SN and the solid blue lines represent he distances between SNs. The dotted blue lines connected to SN 6 represent a change in the respective distances if the 6th SN is shifted from its up to lower position. (c), Graphical representation of connecting adjacent SN in the nearest SN ranking when more that one SN lives in the xz-plane (i.e. has only one red point where it can live). In this case moving the 6th SN from the top red point to the bottom red point leaves the distances between its adjacent SNs unchanged.

For any SN configuration which does not have an ambiguity in the nearest SN ranking, a descriptor vector with a set of ordered distances will uniquely describe the SN configuration up to a rotation and reflection. Before moving on there are two important notes about the descriptor which only uses distances. First, the descriptor constructed in this work using only distances works because the strict ordering of distances keeps the reference to which pair of ions a given distance belongs. In a case when the distances are binned into histograms this additional information is lost. As such even with all distances given, reconstructing a single configuration of ions may not be possible. Second, by including all distance between ions in the descriptor vector, it is clear for SN configurations such as the one shown in Fig. 1 (b) the descriptor vector will contain a large amount of redundant information. Work is ongoing to improve this.

III. DFT computational details

T (kK)	N	ρ (g/cm^3)	# bands	E cutoff (Ry)	k-points (KS)	Real space grid (OF)	Δt (fs)
10	20	1.0	36	600	6x6x6	64x72x72	0.24
30	20	1.0	64	700	6x6x6	64x72x72	0.24
60	20	1.0	80	750	6x6x6	64x64x64	0.12
90	20	1.0	80	750	6x6x6	64x64x64	0.12
90	20	5.0	100	750	6x6x6	32x64x64	0.12
90	540	1.0	1200	500	1x1x1	256x256x256	0.12
150	20	1.0	128	800	6x6x6	128x128x128	0.048

Table 1: Parameters of the DFT calculations for quick reference.

All DFT calculations were performed with the PROFESS@Q-ESPRESSO package [1-5]. Found in table 1 columns 4-6 are the values of the technical parameters used in the KS calculations at the various thermodynamic conditions considered. Column 7 indicates the real space grid for the OF calculations. The last column of table 1 indicates the time step for all MD simulations (OF, KS or ML) performed at the given set of thermodynamic conditions.

To determine the technical parameters found in table 1 a series of convergence tests were performed. For the convergence tests, at given set of thermodynamic conditions, a series of single point calculations with varying technical parameters was performed on a single fixed snapshot of hydrogen atoms. The average change in both the force magnitudes and direction obtained from the distribution of N ions was used to determine the best choice of the technical parameters. In all cases, on average the force magnitude and direction were converged within 1%.

IV. ML implementation in QE

At the current time, the force correction model is comprised of two separate sets of code written in python 3.6. The first set, is the code written for the training procedure; the second is the code that is interfaced with Quantum Espresso (QE) for MD simulations. In both cases the python code requires the following two input files, *ML_input* and *ML_path.* The *ML_input* file contains all of the parameters needed to set up the creation of the descriptors as well as set up the neural network (NN). *ML_path* contains the path to the directory where the *ML_input* file can be found.

A. Training procedure code

To generate the reference data, the existing version of the PROFESS@Q-ESPRESSO package has been modified to allow for both a KS and OF calculation to be performed during a single point calculation. To perform such a calculation one must set up the standard *QE.in* file for QE with the new keyword *mlforce* set to *TRUE* (the input for Profess must also be provided here). QE will then write out the file *ML_reference_data* which contains the ion positions and both forces.

Once the reference data has been generated the python code *Force_correction.py* can be run. This code, for the training procedure, takes in the *ML_reference_data* file along with the *ML_input* file to set up and generate the input descriptor vectors and corresponding reference force differences. Note, throughout this work the *ML_reference_data* file for the train/validation sets and the test set were constructed separately and all test data was put in a separate directory to ensure it is never touched by the training process.

With the reference inputs and outputs constructed the python code begins to train the NN. Here, the NN has been written from scratch using just standard linear algebra operations found in the *NumPy* library [6]. The gradient descent algorithm used to determine the optimal free parameters is also written from scratch, the details of which are provided in section IV. Once the training procedure is finished, the python code writes the optimal parameters of the NN to the files *Weights* and *Bias*.

The primary motivation for this stand-alone code for the training process was to provide the user with flexibility to experiment with the training of the model. The goal of future work will be to have an additional code for the training process that is directly interfaced with QE to enable on-the-fly MD simulations.

B. Molecular dynamics code interface

Schematically, the flow of the PROFESS@Q-ESPRESSO package interfaced with the python code for MD simulations is shown in Fig. 2. Starting with the initialization, QE requires all usual input files needed for KS MD to be set up. However, instead of using the keyword *useofdft* to signal that OF calculations will be performed the keywords *mlforce* and *mlfready* must be used in the *QE.in* file, both of which must be set to *TRUE*. Additionally, the standard input files for PROFESS must also be provided. Once QE is initialized it will automatically initialize Profess; this is not the case for the python code. Here an additional job must be submitted to run the python code. The reason for this is QE and the python code are, at the moment, are interfaced in such a way that all data transferred between them is written and then read from a set of temporary files in the working directory.

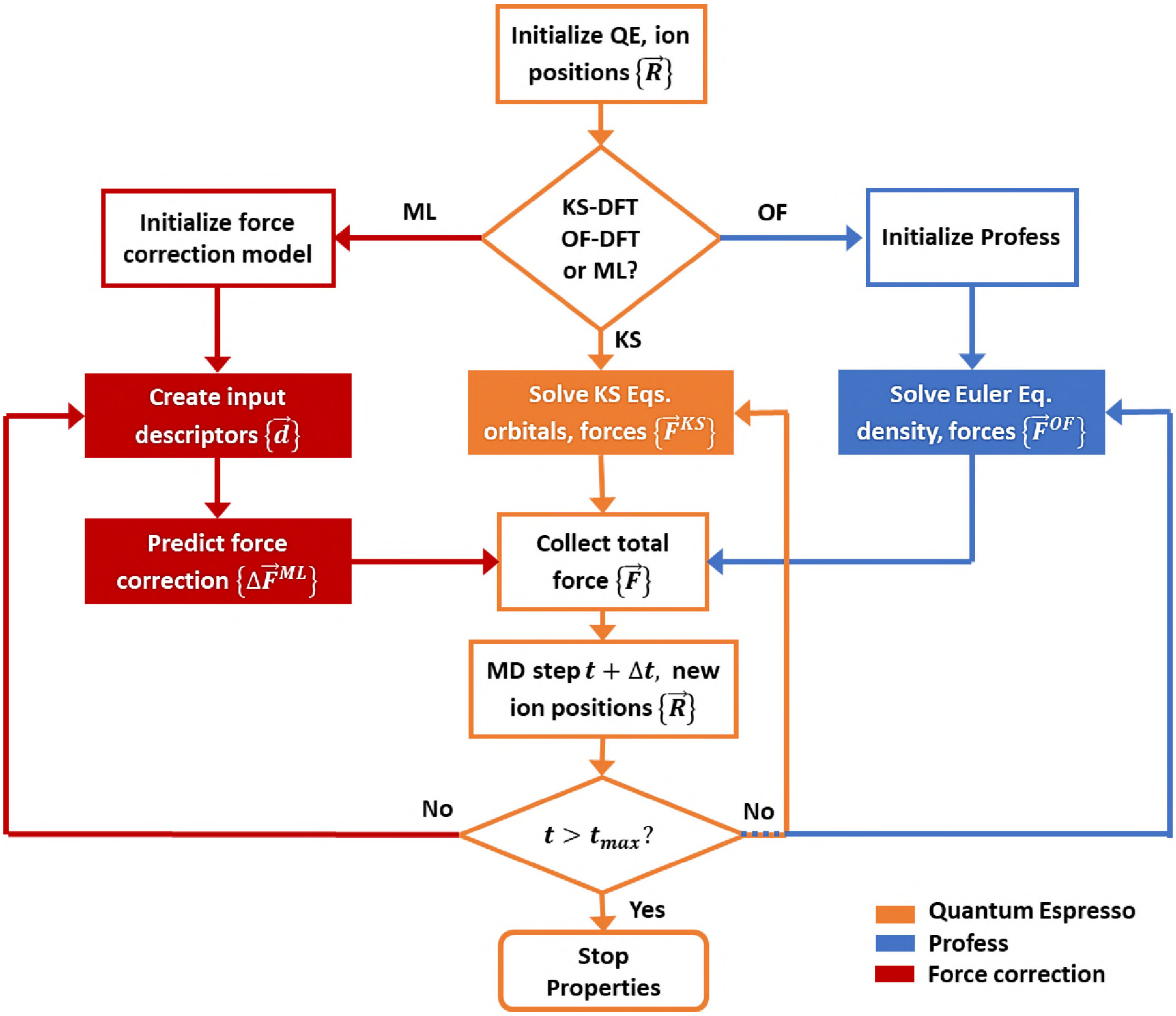

Initialize QE, ion positions $\{\vec{R}\}$
KS-DFT OF-DFT or ML?
ML
OF
KS
Initialize force correction model
Initialize Profess
Create input descriptors $\{\vec{d}\}$
Solve KS Eqs. orbitals, forces $\{\vec{F}^{KS}\}$
Solve Euler Eq. density, forces $\{\vec{F}^{OF}\}$
Predict force correction $\{\Delta\vec{F}^{ML}\}$
Collect total force $\{\vec{F}\}$
MD step $t+\Delta t$, new ion positions $\{\vec{R}\}$
No
No
$t > t_{max}$?
Yes
Stop Properties
Quantum Espresso
Profess
Force correction

Figure 2: Schematic diagram of how the ML based code has been interfaced with the PROFESS@Q-ESPRESSO package.

With all initializations complete, QE passes the ion positions to both Profess and to the python code. Since the python code is run as a separate job, the construction of the descriptors and prediction of the force difference occurs simultaneously with the calculation of the OF forces. Note, in Fig. 2 this corresponds to simultaneously moving from the upper diamond into the red and blue branches; the code for calculating KS forces is not activated. Once the OF forces and corrections have been calculated, QE combines them to produce the equivalent KS forces. These equivalent KS forces are then used to move the ions. This process is continued until the required number of MD steps is reached. The calculation of the energies and pressure for the resulting MD trajectory is carried out as part of post processing.

The above interfacing is done in such a way as to preserve the standard functionality of the PROFESS@Q-ESPRESSO package. That is, with *mlforce, mlfready* and *useofdft* set to *FALSE*, standard KS MD will be performed. This is indicated by the purely orange loop of Fig. 1. If both ML keywords are set to *FALSE* and *useofdft* is set to *TRUE*, OF based MD is performed (blue-orange loop Fig. 1).

It is also of note that all python codes for the force correction model have been parallelized using the mpi4pi library [7-10].

V. Gradient descent

The ML force correction for a given reference ion has the form

$$\Delta \vec{F}_i^{ML} = \sum_j NN(\vec{d}_{ij})\hat{R}_{ij}. \tag{8}$$

Here, a NN is used to map the description, $\vec{d}_{ij}$, of the j^{th} first neighbor (FN) to its contribution to the force difference for the i^{th} reference ion. Moving forward the ML force difference will be written in the form

$$\Delta\vec{F}_i^{ML} = \left(\vec{P}^{(1)}C_i\right) \circ R_i\vec{P}_i. \tag{9}$$

The length of the vectors $\vec{P}^{(1)}$and $\vec{P}_i$ are 3 and the number of FNs respectively. In both cases all the elements of the vector are 1.0. The matrix R_i is formed out of the unit vectors pointing to the FNs,

$$R_i = \left(\hat{R}_{i,1}\,\hat{R}_{i,2}\,\hat{R}_{i,3}\,\ldots\right). \tag{10}$$

The symbol $\circ$ indicates a Hadamard product (elementwise multiplication). Note, the Hadamard product in our notation is given higher priority in the order of operations than standard matrix multiplication but will have a lower priority than parenthesis.

The matrix C_i (which is actually a row vector) is formed out of the predicted FN weights,

$$C_i = W^{(2)}f\left(W^{(1)}D_i + B_i\right). \tag{11}$$

Here, $W^{(1)}$ and $W^{(2)}$ are the weights of the NN and f is the activation function. The bias vector, $\vec{\beta}$, added at the hidden layer is constrained to be exactly the same for all predictions. This leads to the B_i matrix which has the form

$$B_i = \vec{\beta}\vec{P}_i^T. \tag{12}$$

The matrix D_i in Eq. (11) is the input matrix and is constructed out of the individual descriptor vectors for each FN,

$$D_i = \left(\vec{d}_{i,1}\,\vec{d}_{i,2}\,\vec{d}_{i,3}\,\ldots\right). \tag{13}$$

To determine the optimal weights and bias of the NN a gradient descent is performed on a cost surface. For this work the cost is defined as

$$C = \frac{1}{2N_S}\sum_{i}^{N_S}\left\{\left|\delta\vec{F}_i\right|^2 + Aexp\left(-\alpha\left|\Delta\vec{F}_i^{ML}\right|^2\right)\right\}, \tag{14}$$

where

$$\delta\vec{F}_i = \Delta\vec{F}_i^{ML} - \Delta\vec{F}_i^{ref}. \tag{15}$$

The gradients of the cost function in terms of the weights and bias are

$$\frac{\partial C}{\partial W^{(2)}} = \frac{1}{N_S}\sum_{i}^{N_S}\left(\delta\vec{F}_i - 2A\alpha\Delta\vec{F}_i^{ML}exp\left(-\alpha\left|\Delta\vec{F}_i^{ML}\right|^2\right)\right)^T R_i f_i^T, \tag{16}$$

$$\frac{\partial C}{\partial W^{(1)}} = \frac{1}{N_S}\sum_{i}^{N_S} f_i' \circ \left(W^{(2)^T}\left\{\delta\vec{F}_i - 2A\alpha\Delta\vec{F}_i^{ML}exp\left(-\alpha\left|\Delta\vec{F}_i^{ML}\right|^2\right)\right\}^T R_i\right) D_i^T \tag{17}$$

and

$$\frac{\partial C}{\partial \vec{\beta}} = \frac{1}{N_S}\sum_{i}^{N_S} f_i' \circ \left(W^{(2)^T}\left\{\delta\vec{F}_i - 2A\alpha\Delta\vec{F}_i^{ML}exp\left(-\alpha\left|\Delta\vec{F}_i^{ML}\right|^2\right)\right\}^T R_i\right) \vec{P}_i. \tag{18}$$

Note, the use of matrix notation in the derivative should be interpreted as taking the derivative of the cost with respect to each individual element of the corresponding matrix or vector. As short hand notation, the argument of f and f' (derivative of the activation function) has been suppressed as they are consistent with the argument of f in Eq. (11).

With the gradient determined the weights and bias are updated as

$$W_{new}^{(2)} = W_{old}^{(2)} - \mu\frac{\partial C}{\partial W^{(2)}}, \tag{19}$$

$$W_{new}^{(1)} = W_{old}^{(1)} - \mu \frac{\partial C}{\partial W^{(1)}} \tag{20}$$

and

$$\vec{\beta}_{new} = \vec{\beta}_{old} - \mu \frac{\partial C}{\partial \vec{\beta}} \tag{21}$$

with μ being the learning rate. For this work μ is a random variable which will be sample from a uniform distribution defined on a certain range of values.

VI. Training the model

A. ML technical parameters

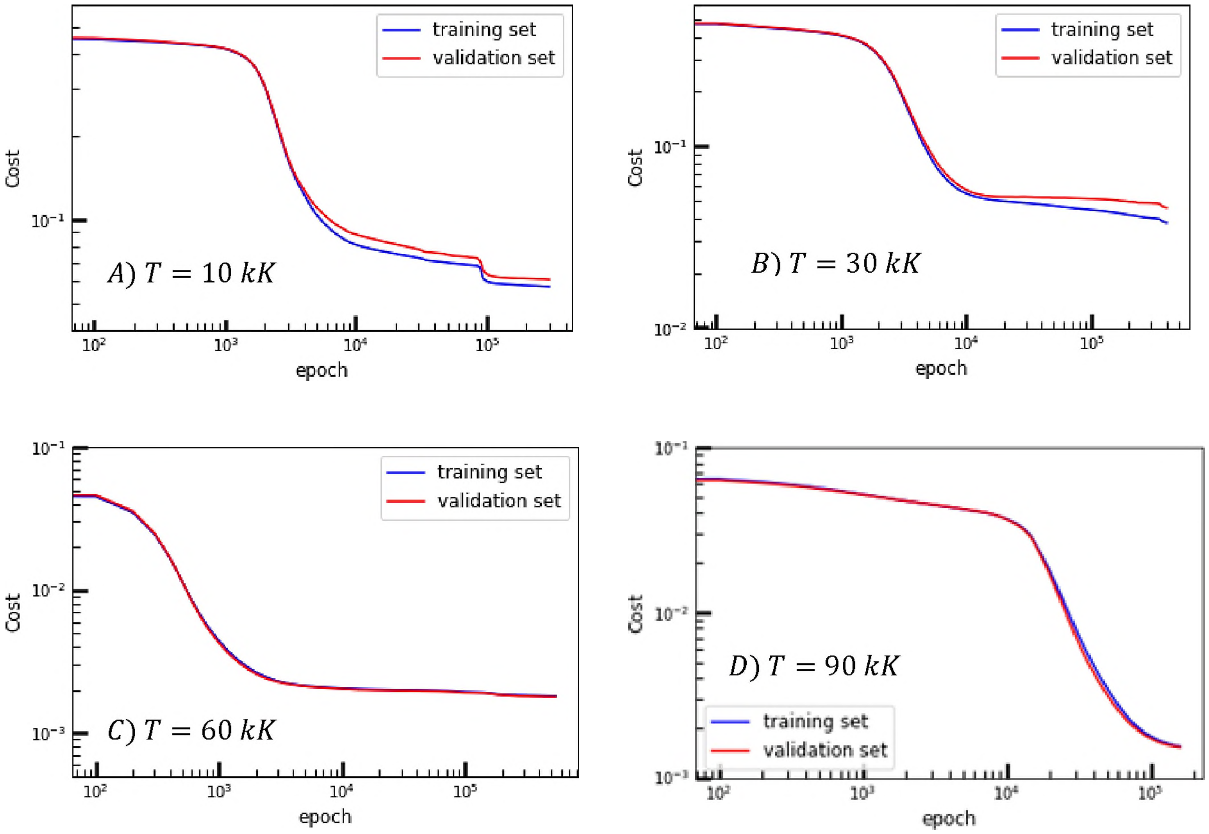
T(kK)	# nodes	# SN	R_c (bohr)	μ	Initialization
10	40	5	6.08	0.001	[-0.1, 0.1]
30	100	5	5.70	0.0008	[-0.1, 0.1]
60	60	3	4.75	0.01	[-0.1, 0.1]
90	40	3	3.78	0.001	[-0.1, 0.1]
150	40	3	3.78	0.005	[-0.1, 0.1]

Table 2: Technical parameters of the training process for quick reference.

Found in table 2 are the technical parameters associated with the creation of the descriptors and the setup of the NN. In all calculations a single layer NN consisting of 40 to 100 nodes is used. The number of nodes along with the cutoff radius, R_c, and the number SNs was determined through a series of convergence tests of the model accuracy.

The parameter μ listed in table 2 is the average value of the learning rate. For all training processes the learning rate is allowed to vary by $\pm 25\%$ of the mean value. The final column of table 2 labeled “initialization” indicates the range that each element of the weight matrices and bias vector are initialized on. Note, for this initialization the elements of the weights and bias are drawn randomly form a uniform distribution defined on the given range.

B. Learning curves:



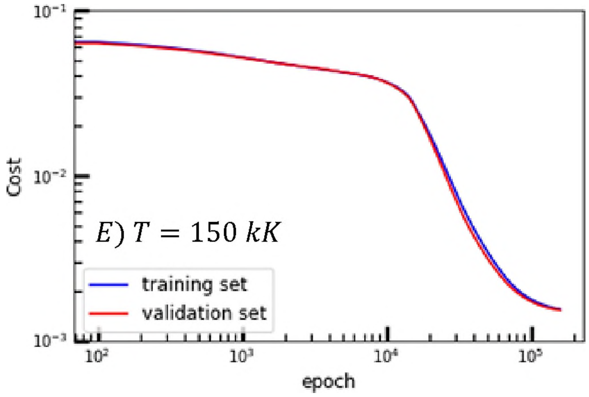


Figure 3: Learning curves for one member of the ensemble of force correction models constructed at each temperature considered.

For each set of thermodynamic conditions considered an ensemble of 15 force correction models was constructed. Shown in Fig. 3 are the learning curves for one randomly chosen member of the ensemble at each temperature. The blue and red curves represent the cost defined in Eq. (14) calculated on the training and validation sets respectively. As can be seen, there is often a period of rapid improvement in the model followed by a plateau. Pushing the training run to 2 million epochs did not show any change in these plateaus. The overlap of the validation and training curves indicates that the level of accuracy achieved on the training set is likely to generalize well to unseen data (whether that level of accuracy is "good" will be measured by the test set in the next subsection). However, it should be noted that the cost plateaus at a value of 0.1 at the lowest temperatures whereas it plateaus at 0.001 for the highest temperatures. This signals that additional complexity in the model is likely needed at the lowest temperatures considered in order to further improve the model predictions.

C. Performance on the test set

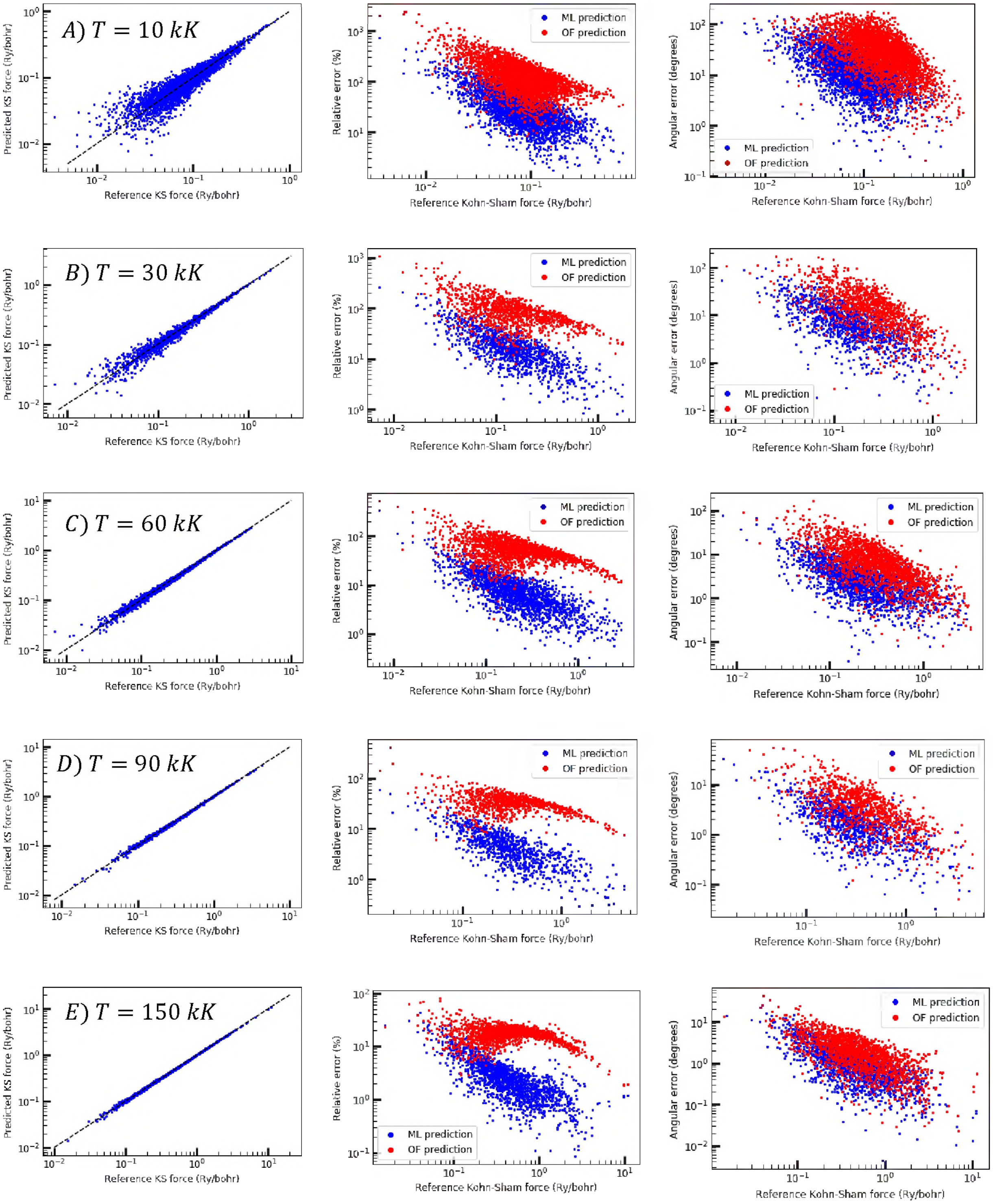
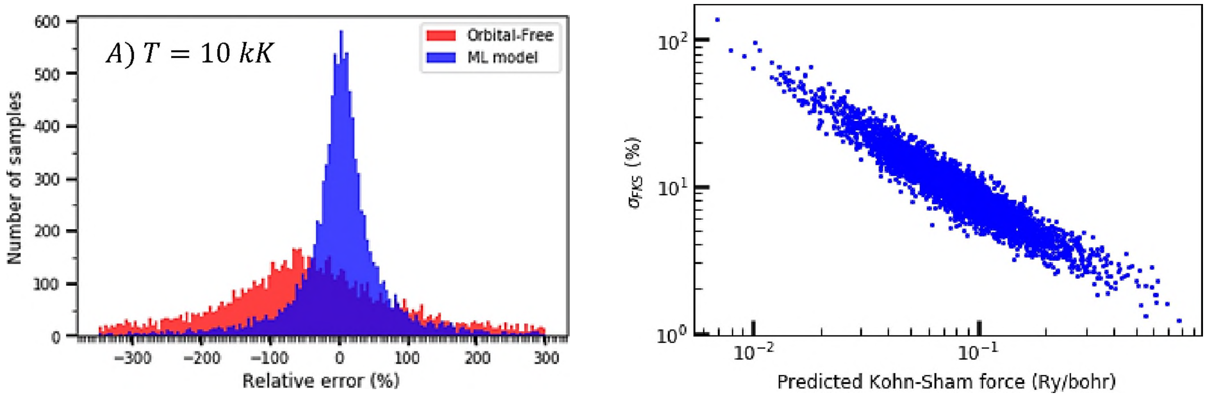


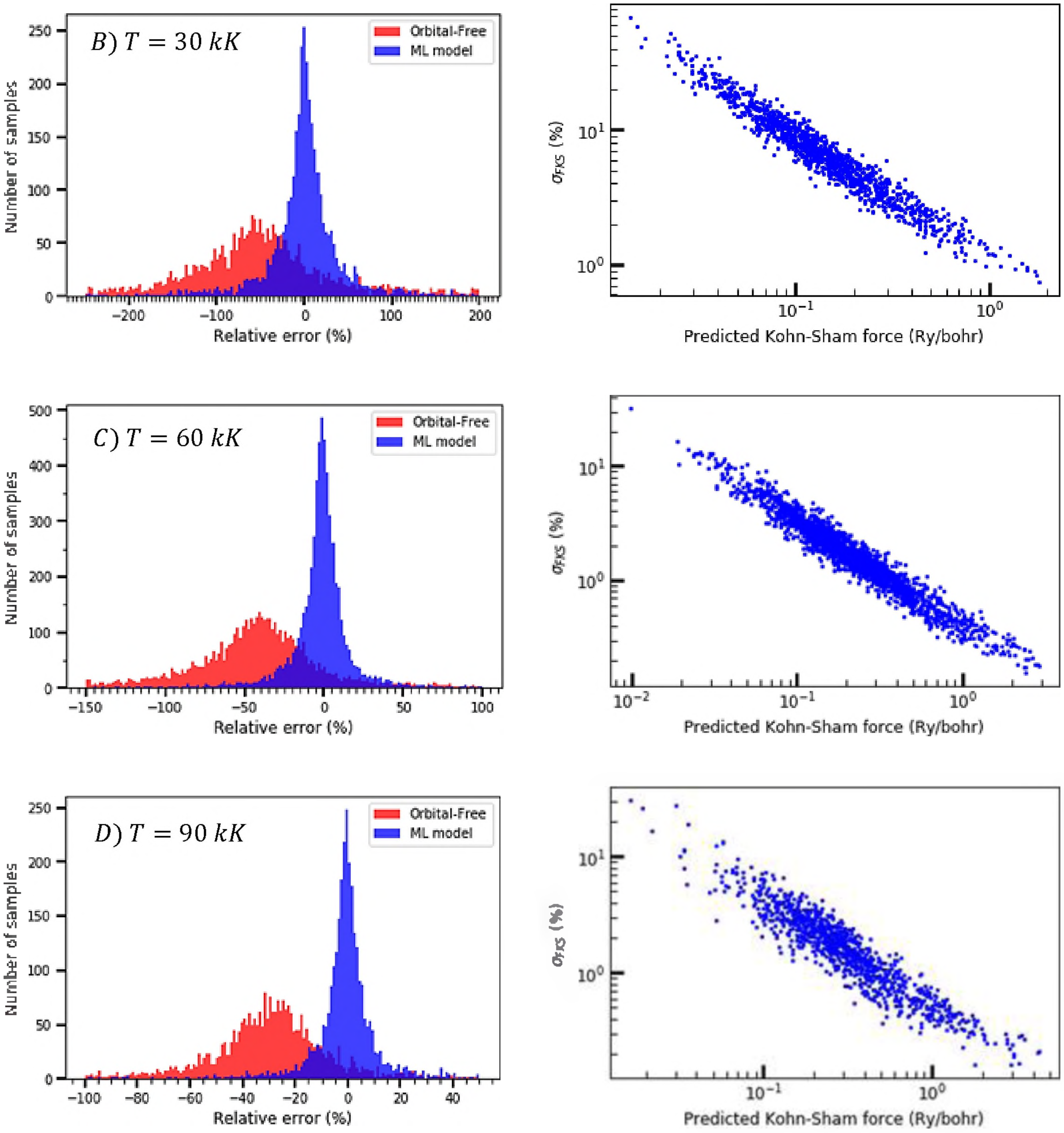
Figure 4: Results on the test set for the force correction model at each temperature considered.

Column 1, predicted KS forces as a function of the target KS forces. The black dotted line represents the target the model aims to achieve. Column 2, the relative errors in the predicted KS force (blue) and OF force (red) magnitudes as a function of the target KS force magnitude. Column 3, Angular deviation of the predicted KS force (blue) and OF force (red) from the target KS force as a function of the magnitude of the target KS force.

Once the ensemble of force correction models are trained at a given temperature, the accuracy of the average force predicted by the ensemble is determined by making predictions on a previously set aside test set. The resulting predictions on the test sets are shown in Fig. 4.

It is clear from Fig. 4 that the overall accuracy of the model improves as the temperature of the system is increased. This is expected as both KS and OF DFT begin to tend towards the same high temperature limit of the uniform electron gas making the correction easier to learn. This leads to a majority of predicted KS forces at 30 kK and above having a magnitude within 10% of the target values and a directional deviation with respect to the target KS force of less than 10 degrees. At all temperatures considered the force correction model is capable of producing a force that is more accurate than the underlying OF force.




$B)\ T = 30\ kK$
Orbital-Free
ML model
Number of samples
0
50
100
150
200
250
−200
−100
0
100
200
Relative error (%)
σ_{FKS} (%)
10^1
10^0
10^{-1}
10^0
Predicted Kohn-Sham force (Ry/bohr)
$C)\ T = 60\ kK$
Orbital-Free
ML model
Number of samples
0
100
200
300
400
500
−150
−100
−50
0
50
100
Relative error (%)
σ_{FKS} (%)
10^1
10^0
10^{-2}
10^{-1}
10^0
Predicted Kohn-Sham force (Ry/bohr)
$D)\ T = 90\ kK$
Orbital-Free
ML model
Number of samples
0
50
100
150
200
250
−100
−80
−60
−40
−20
0
20
40
Relative error (%)
σ_{FKS} (%)
10^1
10^0
10^{-1}
10^0
Predicted Kohn-Sham force (Ry/bohr)

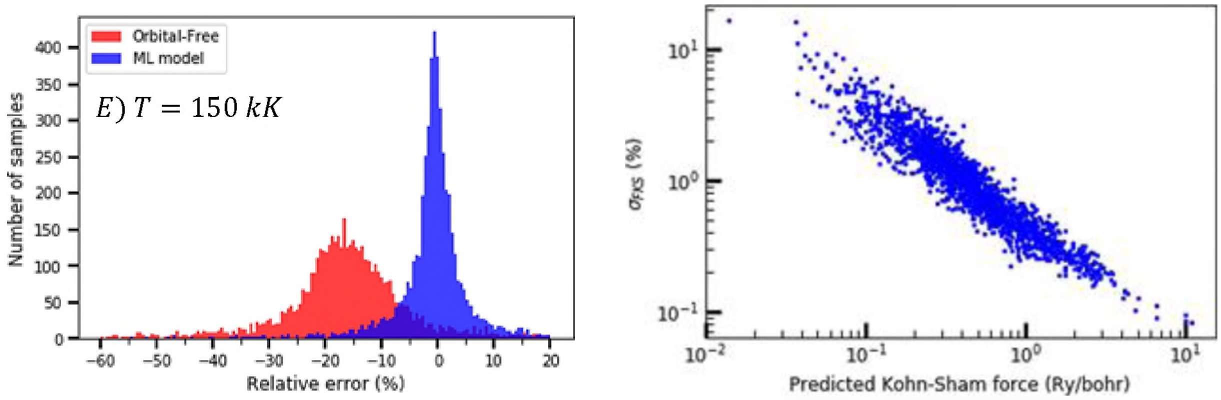


Figure 5: Column 1, relative error distributions for the force components for the predicted KS forces (blue) and OF force (red) for the ions in the test sets at the respective temperature. Column 2, distribution of uncertainties in the predicted KS forces for predictions made on the test set.

The resulting errors on the test sets are further analyzed in column 1 of Fig. 5. Here, it can be seen that the error distribution on the components of the predicted KS forces is symmetric around 0%. This is not the case for the underlying OF forces which show a consistent over estimation in all of the force components. Note, the numerator of the relative error is defined as target KS force component minus approximation force component. Furthermore, as the temperature of interest is decreased, the relative error distribution for the predicted KS force components remains symmetric. This ultimately leads to the symmetric spread in the predicted KS force magnitudes about the target KS force magnitudes observed in Fig. 4.

Also shown in Fig. 5 are the distributions of uncertainties on the predicted KS forces in the test set. Similar to the relative errors, the uncertainties tend to increase as the temperature of the system is decreased. Despite this overall the force correction model tends to make a predicted KS force with a 10% or less uncertainty.

VII. ML based MD

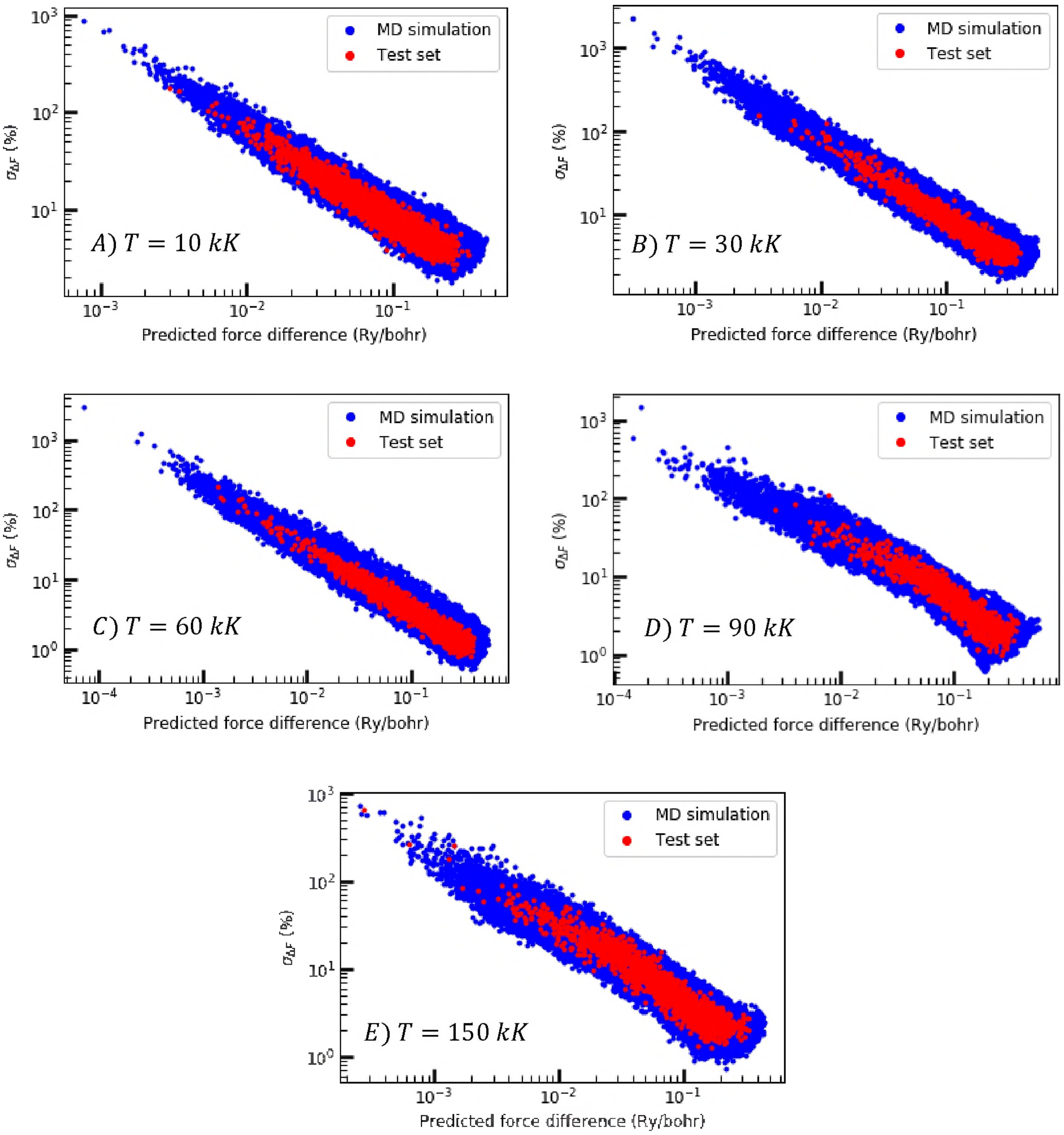


Figure 6: Distribution of uncertainties from the MD dynamic simulation (blue) with the force correction model compared to the distribution of uncertainties from the test set (red).

During the course of MD simulations it is important to confirm that the force correction model stays within the domain of the input vector space associated with the training set to ensure the accuracy of the predictions is maintained. Within this work, this is done by examining the uncertainties resulting from the ensemble of trained models used to predict the force differences. Here, the uncertainty is defined as

$$\sigma_{\Delta F} = \frac{|\vec{\sigma}|}{\left|\overrightarrow{\Delta F}^{ML}\right|} x100\%, \tag{22}$$

where $\vec{\sigma}$ is the vector formed out of the standard deviations in each component of the predicted force difference.

The uncertainty defined by Eq. (22) is different than those in the main text used during the testing of the model. This is because the OF forces were discarded during the MD simulation (an oversight that has been corrected). However, it is expected that the sensitivity of the uncertainty in the force difference to unseen local configurations should lie between that of the predict KS force and the predicted FN weights. Since both of those uncertainties where in agreement during the testing of the model, the uncertainty in the force difference should provide reasonable assessment as to whether unseen local configurations are encountered.

Shown in Fig. 6 are the distributions of the uncertainties in the force differences for the MD trajectory and the test sets for all temperatures considered. As can be seen, the uncertainty distribution from the test set falls directly in the middle of the distribution from the MD simulations for all temperatures. The presence of blue points around the red points can be attributed to higher sampling of the tails of the uncertainty distribution during the MD simulations as the blue distribution contains around 30000 points whereas the red distribution contains no more than 2000 points. This suggests that the MD simulation likely samples regions of the input vector space for which data is presents but the data density is relatively low. However, given that the uncertainty distribution from the MD simulation appears to grow symmetrically around the distribution from the test set (as opposed to the shift in the distribution observed during the transferability tests) it is unlikely that a significant number of new local configurations are encounter during the MD runs. As such the results of the MD simulation should be trusted as was confirmed with the distributions of energies and pressures.

VIII. Behler and Parrinello comparison

The developed force correction model is benchmarked to the Behler and Parrinello (BP) scheme [11] in three different ways. In the first benchmarking the descriptors based on inter-ionic distances describing the SN configuration is swapped out for the BP descriptors (Eqs. (4) and (5) of [11]). The results of the hyperparameter search at 90 kK are shown in table 3 with the corresponding hyperparameters listed in table 4. In comparison two the full BP model is used to predict force difference at 90 kK. See table 5 for the results on the test set and table 6 for the hyperparameters. For the third and final comparison the full BP model is used to directly predict the KS forces. The results on the test set at 90 kK are shown in table 7 with the corresponding hyperparameters listed in table 8.

For each BP based model in the three benchmarkings a single layer fully connect feed-forward NN with 40 nodes in the hidden layer was trained according the corresponding form of the cost in Eq. (14). For comparison three A and α in the exponential regularization were set to 400 and 1700 respectively. For a single model in the hyperparameter search for comparison 2 the mean squared error (MSE) with respect to the energy was added into the cost with equal weight to the MSE in the forces. This addition was shown to decrease the accuracy in the resulting force predictions, table 5 row 1. As such, only force data is used in the training of all other BP models to help ensure the highest level of accuracy in the predicted forces.

For all BP models in the comparison, the activation function, master training set and test set are the same as those used in the training of the force correction model developed here. For each of the BP models the learning rate was on the order of 10^{-4} with typical training runs taking 50 to 100 k epochs to optimize the NN. The learning curves for the models in table 3 row 4, table 5 row 2 and table 7 row 2 are shown in Fig. 7.

Des.	**SN cutoff (bohr)**	**Rel. Err. (%)**	**25th**	**50th**	**75th**	**Ang. Err. (degrees)**	**25th**	**50th**	**75th**
1 rad. 2 ang.	1.89	19.77	8.76	13.33	22.07	7.81	2.65	4.95	8.90
2 rad. 4 ang.	2.83	10.79	5.43	7.72	12.74	3.72	1.16	2.22	4.23
4 rad. 4 ang.	2.83	10.32	3.76	6.50	10.94	4.34	1.20	2.39	4.39
6 rad. 4 ang.	3.78	8.19	4.80	6.82	9.92	3.08	1.26	2.34	3.83
12 rad. 8 ang.	3.78	9.75	5.47	8.29	11.85	3.49	1.39	2.56	4.43

Table 3: KS predictions on the 90 kK test set for the force correction model that utilizes the Behler and Parrinello descriptors to describe the SN configurations.

Des.	**SN cutoff (bohr)**	**Radial descriptor parameters** (η, R_s)	**Angular descriptor parameters** (η, ζ, λ)
1 rad. 2 ang.	1.89	(0.28, 0.378)	(0.56, 1.0, ±1)
2 rad. 4 ang.	2.83	(0.28, 0.0), (5.61, 1.51)	(0.28, 1.4, ±1), (0.28, 4.5, ±1)
4 rad. 4 ang.	2.83	(0.28, 0.0), (1.40, 0.47), (4.20,0.94), (8.2, 1.89)	(0.28, 1.4, ±1), (0.28, 4.5, ±1)

6 rad. 4 ang.	3.78	(0.28, 0.0), (1.40,0.57), (4.20, 1.13), (8.2, 1.70), (12.61, 2.27), (16.82, 2.83)	(0.28, 1.4, ±1), (0.28, 4.5, ±1)
12 rad. 8 ang.	3.78	(0.28, 0.0), (1.12, 0.38), (1.68, 0.76), (4.2, 0.94), (5.61, 1.13), (8.41, 1.42), (8.41, 1.7), (8.41, 1.98), (8.41, 2.27), (8.41, 2.55), (8.41, 2.83), (8.41, 3.31)	(0.28, 1.4, ±1), (0.28, 2.5, ±1), (0.28, 4.0, ±1), (0.28, 7.0, ±1)

Table 4: Hyperparameters of the descriptors used in the calculations of table 3. Note: the units of the hyperparameters are η (1/bohr^2), R_s (bohr), ζ and λ unitless.

Des.	**Rel. Err. (%)**	**25th**	**50th**	**75th**	**Ang. Err. (degrees)**	**25th**	**50th**	**75th**
2 rad. 4 ang. (0.5, 0.5)	22.01	13.20	19.35	27.18	5.16	1.92	3.65	6.29
2 rad. 4 ang.	12.23	5.75	9.52	15.88	4.07	1.48	2.90	5.04
10 rad. 10 ang.	16.23	8.07	13.10	20.29	5.54	2.06	3.87	6.83
20 rad. 22 ang.	23.90	12.52	20.48	30.29	6.19	2.35	4.37	7.54

Table 5: KS predictions on the 90 kK test set for the full BP model trained on force differences. The (0.5, 0.5) in row 1 column 1 indicates the mean squared error in energies and forces have the same weight in the cost function. All other runs use solely force data for training.

Des.	**Radial descriptor parameters (η, R_s)**	**Angular descriptor parameters (η, ζ, λ)**

2 rad. 4 ang. **(0.5, 0.5)**	(0.0, 1.0), (1.0, 1.89)	(0.2, 1.5, ±1), (0.2, 6.0, ±1)
2 rad. 4 ang.	(0.0, 1.0), (1.0, 1.89)	(0.2, 1.5, ±1), (0.2, 6.0, ±1)
10 rad. 10 ang.	(20, 0.0), (20, 0.38), (20, 0.76), (20, 1.13), (20, 1.51), (20, 1.89), (20, 2.27), (20, 2.65), (20, 3.02), (20, 3.40)	(0.2, 1.0, ±1), (0.2, 2.5, ±1), (0.2, 4.0, ±1), (0.2, 5.5, ±1), (0.2, 9.0, ±1)
20 rad. 22 ang.	(80.0, 0.0), (80.0, 0.19), (80.0, 0.38), (80.0, 0.57), (80.0, 0.76), (80.0, 0.94), (80.0, 1.13), (80.0, 1.32), (80.0, 1.51), (80.0, 1.7), (80.0, 1.89), (80.0, 2.08), (80.0, 2.27), (80.0, 2.46), (80.0, 2.65), (80.0, 2.83), (80.0, 3.02), (80.0, 3.21), (80.0, 3.4), (80.0, 3.59)	(0.2, 1.0, ±1), (0.2, 1.5, ±1), (0.2, 2.0, ±1), (0.2, 3.5, ±1), (0.2, 5.0, ±1), (0.2, 6.5, ±1), (0.2, 8.0, ±1), (0.2, 10.0, ±1), (0.2, 18.0, ±1), (0.2, 25.0, ±1), (0.2, 36.0, ±1)

Table 6: Hyperparameters of the descriptors used in the calculations of table 5.

Note: the units of the hyperparameters are η (1/bohr^2), R_s (bohr), ζ and λ unitless.

Des.	**R_c (bohr)**	**Rel. Err. (%)**	**25th**	**50th**	**75th**	**Ang. Err. (degrees)**	**25th**	**50th**	**75th**
29 rad. 24 ang.	5.67	149.41	74.17	112.34	177.11	67.13	34.57	62.00	94.43
30 rad. 24 ang.	9.45	154.52	82.52	120.73	187.98	75.46	42.78	70.85	110.08
40 rad. 24 ang.	9.45	157.87	87.62	121.74	185.42	80.47	46.93	79.28	112.50
40 rad. 24 ang.	14.18	158.17	85.78	120.54	183.60	78.46	44.46	75.62	111.60
40 rad. 24 ang.	17.00	151.19	87.14	117.89	174.25	78.61	47.41	76.27	108.03

Table 7: KS predictions on the 90 kK test set with the full BP model trained on KS forces.

Des.	R_c **(bohr)**	**Rad.** η **(bohr^{-2})**	R_s **(bohr)**	**Ang.** η **(bohr^{-2})**	ζ
29 rad. 24 ang.	5.67	80.0	i$(R_c - 0.3)$/29 i = 0, 1, 2, …, 28	3.0, 0.7, 0.2, 0.05,	1.0, 3.0, 10.0
30 rad. 24 ang.	9.45	40.0	iR_c/30 i = 0, 1, 2, …, 29	1.0, 0.2, 0.05, 0.01	1.0, 3.0, 10.0
40 rad. 24 ang.	9.45	80	iR_c/40 i = 0, 1, 2, …, 39	1.0, 0.2, 0.05, 0.01	1.0, 3.0, 10.0
40 rad. 24 ang.	14.18	40	iR_c/40 i = 0, 1, 2, …, 39	1.0, 0.2, 0.05, 0.01	0.5, 2.0, 10.0
40 rad. 24 ang.	17.00	35	iR_c/40 i = 0, 1, 2, …, 39	0.8, 0.1, 0.02, 0.003	0.5, 2.0, 10.0

Table 8: Hyperparameters of the descriptors used in the calculations of table 7. Note: unlike previous tables all combinations of hyperparameters are used to construct the descriptor vectors. In the case of the angular descriptors λ takes the values ± 1.

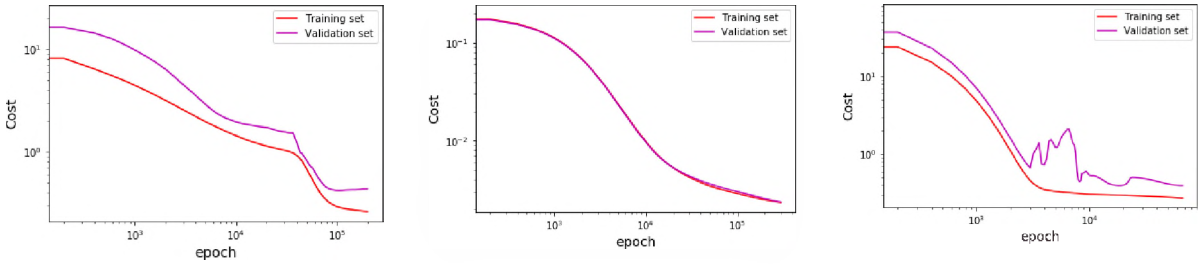


Figure 7: Learning curves for three separate BP based models. (left) 6 rad. 4 ang. model from table 3 row 4, (middle) 2 rad. 4 ang. model from table 5 row 2 and (right) 30 rad. 24 ang. model from table 7 row 2.

Section IV. Assessing the non-conservative nature

As the force correction model predicts forces directly, errors in the predictions will result in a force field that is non-conservative. To assess the non-conservative nature of the force correction model, the first step was to calculate the curl of the predicted KS forces. As the underlying OF forces are conservative the curl of the predicted KS forces will be equal to the curl of the predicted force differences.

The curl of the predicted force was calculated numerically as follows. For a given local configuration the reference ion is perturbed in a specific direction while keeping the surrounding FNs fixed. The new descriptor vector is calculated for the perturbed reference ion and the corresponding force difference is predicted. In an attempt to avoid noise in the derivatives, the reference ion is perturbed four times with a maximum perturbation of ± 0.01 bohr in a given direction. Each force component, as a function of the change in the reference ion position, is fit with a line using a least squares fit and the corresponding slope is taken as the derivative of that force component with respect to the direction of the perturbation. The resulting calculated curl of the predicted KS forces on the 90 kK test set are shown in Fig. 8.

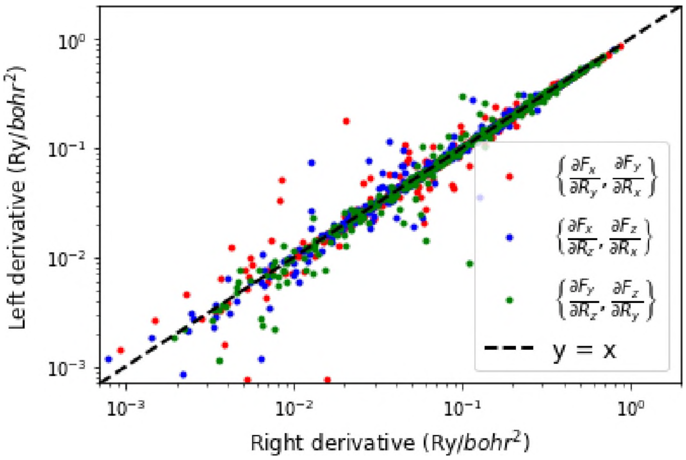


Figure 8: Calculated curl of the predicted KS forces on the 90 kK test set. Right and left derivative indicate the derivative in a given set of brackets in the plot key.

To determine how the non-conservative nature of the predicted forces effects a MD simulation a separate MD simulation was performed to monitor the change in the potential energy of the system from one MD step to the next. Here, the change in potential energy can be calculated as

$$E^{(j+1)} - E^{(j)} = -\Delta t \sum_{i}^{N} \vec{F}_i^{(j)} \cdot \vec{v}_i^{(j)}, \tag{23}$$

where $\vec{F}_i^{(j)}$ is the force on the ith ion at MD step j, $\vec{v}_i^{(j)}$ is the corresponding ionic velocity and Δt is the time step of the simulation. During the course of the MD where the change of energy is observed the thermostat is turned off and the time step is reduce by two orders of magnitude. The latter is done to minimize the non-conservative nature of the Verlet algorithm. This test was then repeated in the same manner for a MD simulation driven by only the OF ionic forces. The results at 90 kK can be seen in Fig. 9. Note, this test is analogous to that performed in [12].

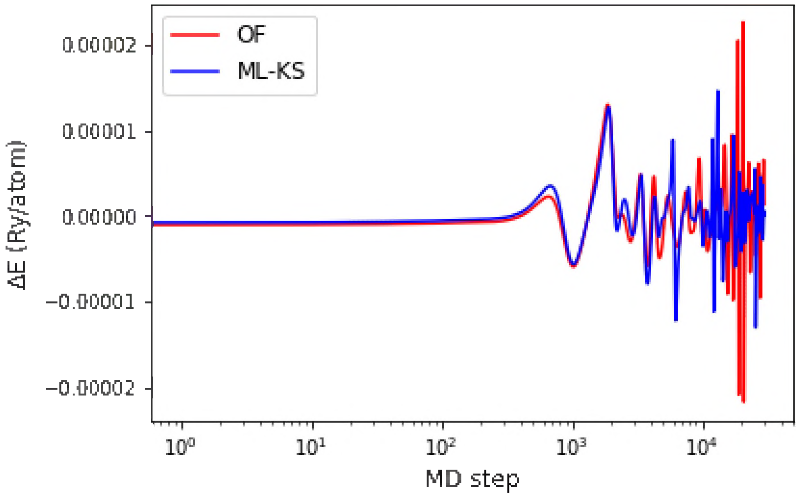


Figure 9: Change in potential energy from one MD step to the next for the force correction model trained at 90 kK (blue) and for uncorrected OF forces (red). The time step for these simulations was 0.00097 fs and both simulation are initialized with the same ionic positions.

References:

1. V. V. Karasiev, T. Sjostrom and S. B. Trickey, Finite-temperature orbital-free DFT molecular dynamics: Coupling PROFESS and QUANTUM ESPRESSO, Comput. Phys. Commun. 185, 12, 3240-3249 (2014).

2. P. Giannozzi, S. Baroni, B. Nicola, N. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, G. L. Chiarotti, M. Cococcioni, I. Dabo et al., QUANTUM ESPRESSO: A modular and open source software project for quantum simulations of materials, J. Phys.: Condens. Matter 21, 395502 (2009).

3. P. Giannozzi, O. Andreussi, T. Brumme, O. Bunau, M. Buongiorno Nardelli, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, M. Cococcioni et al., Advanced capabilities for materials modelling with Quantum ESPRESSO, J. Phys.: Condens. Matter 29, 465901 (2017).

4. G. S. Ho, V. L. Ligneres and E. A. Carter, Introducing PROFESS: A new program for orbital-free density functional theory calculations, Comput. Phys. Commun. **179**, 11, 839-854 (2008).

5. M. Chen, J. Xia, C. Huang, J. M. Dieterich, L. Hung, I. Shin, and E. A. Carter, Introducing PROFESS 3.0: An advanced program for orbital-free density functional theory molecular dynamics simulations, Comput. Phys. Commun. 190, 228 (2015).

6. C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith et al., Array programming with NumPy, Nature 585, 357 (2020).

7. L. Dalcin, R. Paz, and M. Storti, MPI for Python, Journal of Parallel and Distributed Computing, **65**(9):1108-1115, 2005.

8. L. Dalcin, R. Paz, M. Storti, and J. D'Elia, MPI for Python: performance improvements and MPI-2 extensions, Journal of Parallel and Distributed Computing, **68**(5):655-662, 2008.

9. L. Dalcin, P. Kler, R. Paz, and A. Cosimo, Parallel Distributed Computing using Python, Advances in Water Resources, **34**(9):1124-1139, 2011.

10. L. Dalcin and Y.-L. L. Fang, mpi4py: Status Update After 12 Years of Development, Computing in Science & Engineering, **23**(4):47-54, 2021.

11. J. Behler and M. Parrinello, Generalized neural-network representation of high-dimensional potential-energy surfaces, Phys. Rev. Lett. **98**, 146401 (2007).

12. V. Botu, R. Batra, J. Chapman and R. Ramprasad, Machine Learning Force Fields: Construction, Validation, and Outlook, J. Phys. Chem. C **121**, 511-522 (2017).