Machine learning a highly accurate exchange and correlation functional of the electronic density

Sebastian Dick and Marivi Fernandez-Serra
Physics and Astronomy Department, Stony Brook University, Stony Brook, New York 11794-3800, United States and
Institute for Advanced Computational Science, Stony Brook University, Stony Brook, New York 11794-3800, United States

Density Functional Theory (DFT) is the standard formalism to study the electronic structure of matter at the atomic scale. The balance between accuracy and computational cost that DFT-based simulations provide allows researchers to understand the structural and dynamical properties of increasingly large and complex systems at the quantum mechanical level. In Kohn-Sham DFT, this balance depends on the choice of exchange and correlation functional, which only exists in approximate form. Increasing the non-locality of this functional and climbing the figurative Jacob's ladder of DFT, one can systematically reduce the amount of approximation involved and thus approach the exact functional. Doing this, however, comes at the price of increased computational cost, and so, for extensive systems, the predominant methods of choice can still be found within the lower-rung approximations. Here we propose a framework to create highly accurate density functionals by using supervised machine learning, termed NeuralXC. These machine-learned functionals are designed to lift the accuracy of local and semilocal functionals to that provided by more accurate methods while maintaining their efficiency. We show that the functionals learn a meaningful representation of the physical information contained in the training data, making them transferable across systems. We further demonstrate how a functional optimized on water can reproduce experimental results when used in molecular dynamics simulations. Finally, we discuss the effects that our method has on self-consistent electron densities by comparing these densities to benchmark coupled-cluster results.

For many years, density functional theory (DFT) has served as the standard tool to study the electronic structure of materials and condensed systems. Striking an optimal balance between accuracy and computational cost [1], DFT makes a first-principles description of complex and large systems possible that is otherwise out of reach for more accurate *ab initio* approaches. To achieve this balance, DFT is mapped onto a mean-field single electron description within the Kohn-Sham (KS)[2] approach. In KS-DFT, all the complexities of the many-body electron-electron interaction are reduced within a functional of the density. This functional consists of an exchange (X) and a correlation (C) part, the former capturing effects from Pauli-exchange, and the latter approximating correlations of electrons within the many-body wavefunction.

There is a well-defined roadmap to creating more accurate XC functional formulations, the so-called Jacob's ladder of John Perdew [3, 4], with each rung representing increasing levels of complexity and decreasing levels of approximation to the exact XC functional. Following this roadmap, the simplest approximation to the XC density functional is the local density approximation (LDA) [5]. The next rung, known as the generalized gradient approximations (GGAs) [6], adds a functional dependence on the gradient of the density. In meta-GGA (MGGA) functionals, the kinetic energy density and possibly the Laplacian of the density are introduced in the parameterization of the functional. The construction of functionals following this map allows to incorporate the added complexities in a controlled and physically motivated way, imposing the necessary constraints that these formulations should satisfy to correctly and universally describe the underlying physics. As an example of the success of this approach, the MGGA functional SCAN [7] is considered to be one of the most accurate and efficient methods to simulate both solids [8] and molecular systems [9]. Hybrid functionals move one step closer to the exact solution by using a fixed fraction of Hartree-Fock exchange. Including this "exact exchange" particularly helps correct the well known band-gap problem [10] that local (LDA) and semilocal (GGA) KS-DFT exhibits. Both hybrid and MGGA functionals no longer explicitly depend on the electronic density. Therefore, their corresponding XC potentials are not computed as functional derivatives of the density, but as derivatives of the KS wavefunctions instead [10], and hence are non-multiplicative (each orbital experiences a different XC potential), and more expensive to compute.

A completely different approach to obtaining more accurate functionals is to replace the physically motivated path by a data-driven search. Functionals created following this approach are often referred to as semiempirical [11], and versions of these functionals implement all the previously described levels of approximations. In recent years, unprecedented computational capacity has made the calculation of physical properties of molecules and solids with *ab initio* fully correlated accuracy possible. Such developments have allowed researchers to take the semiempirical approach to the extreme, inaugurating an era of machine learning (ML) methods in density functional development. This path produced the recent ωB97M-V [12], a range-separated hybrid meta-GGA with non-local correlation. It was designed using a combinatorial technique taking Becke's B97 family of semiempirical

functionals [13], augmented with hybrid and non-local correlation components as primary ingredients. The fit was done using a database of accurate single-point calculations on a few thousand molecules. Similarly, using a simple mathematical formulation coined data projection on the parameter subspace (DPPS), Fritz *et al.*[14] showed that it was possible to optimize a GGA functional with non-local correlations for liquid water. This functional was fitted to highly accurate data from coupled-cluster calculations that was also used to optimize the water force field MB-POL[15].

While these latter functionals can already be considered to belong to the machine learning (ML) family, modern ML regression methods make use of algorithms such as artificial neural networks (ANN), kernel ridge regression (KRR) and gaussian process regression (GPR). Grifasi et al. [16] have shown that the electron density for small hydrocarbons can be directly predicted from structural information and Fabrizio et al. [17] have been able to extend this work to non-covalently bonded systems. Chandrasekaran et al. were able to achieve the same goal for solid-state systems by introducing a grid-based structure to electron density mapping using an ANN. Both approaches show great promise to significantly speed up ab-initio calculations as they completely circumvent solving the cubic-scaling self-consistent field (SCF) equations. Other works, including the one presented here, have attempted to parametrize an xc-functional with ML, and we will discuss related methods in detail, below.

In this manuscript, we propose a pathway to construct fully machine-learned functionals that depend explicitly on the electronic density. Our method is an evolution of our recent work [18], in which we proposed a method to correct XC density functionals by learning from the density. This method, which we called machine-learned correcting functionals (MLCF) allowed us to obtain accurate total energies from a converged electronic density obtained from a KS DFT calculation. Building on it, in this manuscript, we show that it is possible to obtain the functional derivatives of MLCFs. We continue by demonstrating that the XC potential thus obtained can be used in SCF total energy KS calculations, hence obtaining fully self-consistent semilocal ML KS density functionals. We call this overall method NeuralXC. We show that these functionals encode meaningful chemical information that extends beyond the training set, hence making the functionals very transferable. In addition, the resulting self-consistent densities are shown to approach the "exact" (at the CCSD(T) level) densities in certain cases, despite not using the density as a target in the training process. The paper is structured as follows: in section (i), we compare our method to other relevant approaches that share ideas similar to ours, section (ii) outlines the theory and methods behind NeuralXC, iii) presents results regarding accuracy and transferability and section iv) summarizes our findings and provides an outlook on future work.

I. RELATED WORK

Bogojeski et al. [19] construct a density functional on top of a reasonably cheap baseline DFT calculation (GGA) that can achieve accuracies close to coupled-cluster results. The main difference between our approaches lies in the choice of basis functions, and the way symmetries are encoded. In their method, the molecule is first aligned with a global coordinate system, which is defined through some molecular axes. The electron density is subsequently expanded in a Fourier basis. These design choices seem to restrict their method to systems of fixed size and limit its transferability. Furthermore, instead of obtaining a potential from the energy regressor, an energy correction is added to the baseline results, similar to earlier work by the authors [18]. Thus, to compute forces, their method relies on an auxiliary model that predicts the electron density, whereas we can directly calculate forces using the Hellmann-Feynman [20] theorem.

Nagai et al. [21] propose a more traditional approach of optimizing the functional form of the exchange-correlation (xc) energy. Being defined on a grid without the need for an additional basis set, it is similar in its form to approaches such as DPPS [14], the main difference being the use of a neural network to flexibly parametrize the functional. The neural network is trained by alternating Monte-Carlo updates on the weights with self-consistent calculations, rather than back-propagation. This enables the authors to include densities in their loss-function but limits training set sizes to a few small systems.

Lei and Medford [22] propose a similar, grid-based approach that uses Maxwell-Cartesian spherical harmonic kernels to construct features for their machine-learned functional. However, instead of using total energies as response variables, they rely on a spatial decomposition of the xc-energy, allowing them to decouple grid-points during training but limiting their method to instances for which such decomposition is available.

Apart from approaches rooted in DFT, others have proposed wave-function based methods that try to predict post-Hartree-Fock energies. Welborn et al. [23] and Cheng et al. [24] use molecular-orbital-based machine learning to predict MP2 and coupled-cluster correlation energies with GPR. Nudejima [25] et al. use grid-based descriptors to learn a regression model that can predict the CCSD(T) correlation energy density using densities obtained from Hartree-Fock calculations as input.

To our knowledge, none of the above mentioned methods, except for that of Nagai et al., is used in self-consistent calculations.

II. METHODS

Charge density representation. The charge density is represented following our earlier work [18] by projecting it onto a set of atom-centered basis functions. Throughout this work the inner cutoff radius was set to

zero, resulting in radial basis functions defined as

$$\tilde{\zeta}_n(r) = \begin{cases} \frac{1}{N} r^2 (r_o - r)^{n+2} & \text{for } r < r_o \\ 0 & \text{else} \end{cases} \quad (1)$$

with an outer cutoff radius r_o and a normalization factor N. The full basis is then given as $\psi_{nlm}(\vec{r}) = Y_l^m(\theta, \phi)\zeta_n(r)$, where $Y_l^m(\theta, \phi)$ are spherical harmonics and ζ_n the orthogonalized radial basis functions (for details see Ref. 18).

The descriptors $c_{nlm}^{\alpha,I}$ for atom I of species α at position $\vec{r}_I$ are obtained by projecting the electron density ρ onto the corresponding basis functions ψ_{nlm}^{α}.

$$c_{nlm}^{\alpha,I} = \int_{\vec{r}} \rho(\vec{r} - \vec{r}_{\alpha,I}) \psi_{nlm}^{\alpha}(\vec{r}). \quad (2)$$

We found it beneficial for certain models to use the modified electron density $\delta\rho$ instead of ρ in Eq. 2. This $\delta\rho$ is defined as the difference between the full electron density and atomic electron density ρ_{atm} the latter being constructed by filling the basis functions with appropriate valence charges (see Ref. 26 for details):

$$\delta\rho(\vec{r}) = \rho(\vec{r}) - \rho_{atm}(\vec{r}) \quad (3)$$

Using this 'neutral' density has the advantage that it is generally smoother than ρ, as peaks around the ion cores cancel out. Moreover, $\delta\rho$ always integrates to zero, regardless of the atomic species involved, suggesting that models trained on it will show better transferability across chemical environments. We have used $\delta\rho$ in all models introduced below except for the one trained on water clusters. Here, cross-validation has determined ρ to produce lower generalization errors.

To avoid erroneous behavior during deployment, the model must respect all physical symmetries. These symmetries include permutation of atoms of the same species, rotations, and reflections. One way to achieve invariance is by data augmentation. During this procedure, symmetry operations are repeatedly applied to the original dataset, and the resulting data is included during training. The idea is that by providing the ML model with invariant data, the underlying symmetries are learned automatically. However, especially in regions in feature space where data is sparse, the model is not guaranteed to be invariant. We opted to solve this problem in two ways: permutational invariance is imposed by the architecture of our neural network as discussed below, whereas rotational invariance and invariance under reflection is encoded in the features themselves.

Starting from our original descriptors c_{nlm}, we can obtain a rotationally invariant version by applying the transformation

$$d_{nl} = \sum_{m=-l}^{l} c_{nlm}^* c_{nlm}. \quad (4)$$

In contrast to our previous work [18] it is not necessary to pick a local coordinate system as the forces are being obtained by taking derivatives of the rotationally invariant energy model and therefore transform covariantly by design.

Machine learned functional. As in our Ref. 18, the permutationally invariant Behler-Parrinello networks [27] were used to parametrize the energy functional. Before passing the symmetrized descriptors $d^{(0)}$ through the neural network, three additional preprocessing steps were employed. First, a variance filter was used, disregarding all features whose variance across the training set was below a threshold value equal to 10^{-10}, effectively de-noising the dataset. Second, the features were projected onto their principal components, only keeping enough components so that an explained variance of γ was achieved, with values of ranging from 0.95 to 1. This step has a regularizing effect and decreases the risk of overfitting. As a final step, all features are scaled so that their values are normally distributed across the training set with zero mean and variance one, a step common in machine learning to ensure fast convergence of the optimization algorithm used to train the neural network. All models were implemented in Tensorflow [28] and trained using the Adam [29] optimizer with training rate $\alpha = 0.001$ and decay rates $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and the sigmoid function was chosen as activation. Hyperparameters such as the learning rate and l2-regularization were determined through cross-validation. However, it should be noted that the final depth (i.e., the number of layers) for each network was dependent on the convergence of the iterative training procedure described below.

Once the energy functional $E_{ML}[n(\vec{r})] = E_{ML}(c[n(\vec{r})])$ has been fitted, the potential V_{ML}, which is required to perform self-consistent calculations, can be obtained through

$$V_{ML}[n(\vec{r})] = \frac{\delta E_{ML}[n]}{\delta n(\vec{r})}. \quad (5)$$

Together with Eq. 2 this translates to

$$V_{ML}[n(\vec{r})] = \sum_{\beta} \frac{\partial E_{ML}}{\partial c_\beta} \frac{\delta c_\beta[n]}{\delta n(\vec{r})} = \sum_{\beta} \frac{\partial E_{ML}}{\partial c_\beta} \psi_\beta(\vec{r}). \quad (6)$$

The resulting potential is therefore a linear combination of the original basis functions, with coefficients being determined by the derivatives of the machine learned energy functional. These derivatives are usually implemented in machine learning software packages and thus straightforward to obtain. The machine learned potential and energy are both added back to their baseline counterparts

$$E_{NXC}[n] = E_{base}[n] + E_{ML}[n] \quad (7)$$

$$V_{NXC}[n] = V_{base}[n] + V_{ML}[n] \quad (8)$$

and the combined functionals (NXC for NeuralXC) can in principle be used in any DFT code. Caution is warranted when calculating forces, as the function has an implicit

dependency on the atomic positions. The force contributions from the machine-learned functional are defined as

$$\frac{\partial E_{ML}}{\partial r_{I,i}} = \sum_{\beta} \frac{\partial E_{ML}}{\partial c_\beta} \frac{\partial c_\beta[n]}{\partial r_{I,i}} = \tag{9}$$

$$\sum_{\beta} \frac{\partial E_{ML}}{\partial c_\beta} \int_{\vec{r}} (\frac{\partial \rho(\vec{r})}{\partial r_{I,i}} \psi_\beta(\vec{r}) + \frac{\partial \psi_\beta(\vec{r})}{\partial r_{I,i}} \rho(\vec{r})) = \tag{10}$$

$$\int_{\vec{r}} \frac{\partial \rho(\vec{r})}{\partial r_{I,i}} V_{ML}(\vec{r}) + \sum_{\beta} \frac{\partial E_{ML}}{\partial c_\beta} \int_{\vec{r}} \frac{\partial \psi_\beta(\vec{r})}{\partial r_{I,i}} \rho(\vec{r}). \tag{11}$$

The first term also appears in standard DFT calculations and is automatically taken care of by recombining the machine learned with the baseline functional. However, the second term, which derives from the fact that the expansion basis was centered around atoms, has to be explicitly calculated.

Iterative training. By altering the xc-functional, the self-consistent electron densities change as well. This fact causes the actual accuracy of the ML functional, defined as the accuracy of the energies and forces obtained by self-consistent calculations with the modified functional, to be lower than the accuracy obtained during the fitting procedure. To remedy this, we employed what we call iterative training: The electron densities and corrected energies obtained with the first iteration of the ML functional $E_{ML}^{(1)}$ are used to train a new iteration which is then in turn used to calculate new densities. This procedure is continued until the accuracy of the obtained functional remains unchanged across two subsequent iterations. The topology of the neural network used in iteration $n+1$ is obtained by freezing the hidden layers of iteration n and adding new hidden layers to the network that are then optimized on the n^{th} iteration of the training densities. This technique is reminiscent of a procedure commonly known as greedy layer-wise training in the deep learning community [30], although with a different goal set.

Datasets. To test the data efficiency of NeuralXC, we made use of a dataset by Chmiela et al. [31], which was created to evaluate the symmetric gradient-domain machine learning (sGDML) force field model. The dataset contains total energies calculated for benzene, toluene, ethanol, and malonaldehyde at the coupled cluster with singles doubles and perturbative triples (CCSD(T)) level with a cc-pVDZ (cc-pVTZ for ethanol) basis set. We further included a set of 1000 water geometries and their associated CCSD(T) total energies calculated with a cc-pVTZ basis set used in [24] and obtainable at [32]. The test sets consist of 500 geometries (1000 for ethanol) whereas the maximum training set size was 1000 for the sGDML data and 500 for water. The reduced training sets were sampled from the full set by employing a k-means clustering algorithm in feature space.

The transferability of our functionals was evaluated by making use of Cheng et al.'s publicly available dataset [32], which contains structures sampled from molecular dynamics simulation at 350 K and their associated total energies calculated at a CCSD(T) level (for details see [24]). We further augmented their dataset with our calculations of 100 structures of ethylene, acetylene, and propene each, all sampled from a 5 ps MD trajectory at 350 K and calculated with CCSD(T) using the cc-pVTZ basis [33]. The calculations were conducted with PySCF [34] using density fitting and the frozen core approximation following the methods employed to create the original dataset by Cheng et al.

The dataset used to train a functional optimized for water contained 400 water monomers, 500 dimers, and 250 trimers. The structures and their corresponding energies were all sampled from the data that was used to fit the MB-pol force-field [15, 35, 36]. In particular, for dimers and trimers, the sampling consisted of two steps: half the samples were obtained by first binning the structures by their corresponding two and three-body energies and then uniformly sampling from these bins. The other half was obtained by randomly sampling the full datasets. This was done to give more weight to the tails of the data distribution and to capture extreme cases which might contain valuable information for our ML model. As the MB-pol dataset only contained dimers and trimers, we randomly sampled monomers from the dimer structures as well. For the energies, we followed Babin et al. [37] and used the highly accurate Partridge-Schwenke potential energy surface for the monomers and the one-body energies of the dimers. Two and three-body energies were extracted from the MB-pol dataset, where CCSD(T) at the complete basis set limit was used to obtain these energies (see Refs. [35, 36] for details).

The s66x8 dataset [38], which contains dissociation curves for 66 non-covalent complexes relevant to biomolecular structures, was used to evaluate the transferability of the water model to heterogeneous systems. Calculations were restricted to a subset of structures that contained at least one water molecule.

The baseline calculations for all of the datasets above were conducted with SIESTA [26] using the PBE [6] exchange-correlation functional with norm-conserving pseudopotentials, a real-space grid cutoff of 400 Ry and a cubic unit cell with lattice constant 30 Å. A doubly-polarized quadruple zeta basis set was used for the water clusters. All other structures were computed with a polarized double zeta basis.

III. RESULTS

Data-efficiency. Frequently, training data is scarce or, as in our case, expensive to obtain. Due to the unfavorable scaling of correlated quantum chemistry methods, the creation of highly accurate datasets for medium to large-sized molecules remains challenging to this day. We would, therefore, like to design a machine learning method that utilizes information contained in the available training data to its full extent.

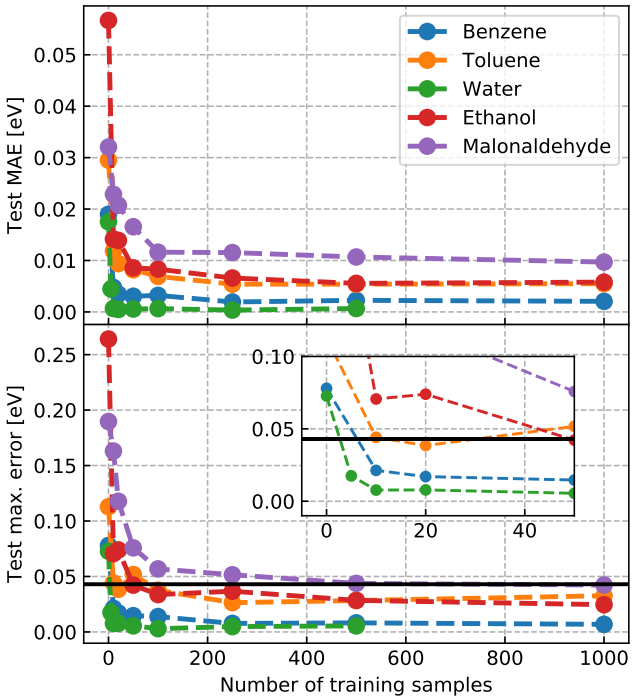

FIG. 1. Training curves for molecules contained in the sGDML [39] dataset. Top: Mean average error in energy prediction on test set with respect to training set size. Bottom: Maximum absolute error in energy prediction on test set. All energy predictions were obtained with self-consistent DFT calculations using NeuralXC as density functional.

Figure 1 shows how the generalization error changes as the size of the training set is increased. For each training set size, a new model was trained from scratch using the iterative approach described above, and self-consistent calculations were run on the entire test set. We used two different metrics for the evaluation: the mean absolute error (MAE) and the maximum absolute error. It can be seen that the MAE saturates at values of 0.01 eV or below at roughly 100 training samples and stays constant beyond. Some improvement in the maximum error can be observed as the training set size is increased further. For malonaldehyde, at least 500 samples are required to reach a max. error below chemical accuracy (1 kcal/mol or 0.043 eV), all other molecules pass that threshold at 100 samples or fewer. It should be noted that the error at which the training curves saturate depends on two factors: the limited expressiveness of the model together with its input representation and sources of noise in the data generation process such as finite grids and supercells. To be more specific, by restricting the model only to use the local electron density expanded in a finite basis set, we limit the amount of information the network can use to infer its energy predictions. Using larger basis sets, and increasing their radial cutoff enables the training curves to converge to smaller errors. However, this will most likely harm data-efficiency and transferability. As we were able to achieve the desired accuracy (max. error below 43 meV) for all molecules, we did not explicitly test the effects of using much larger basis sets.

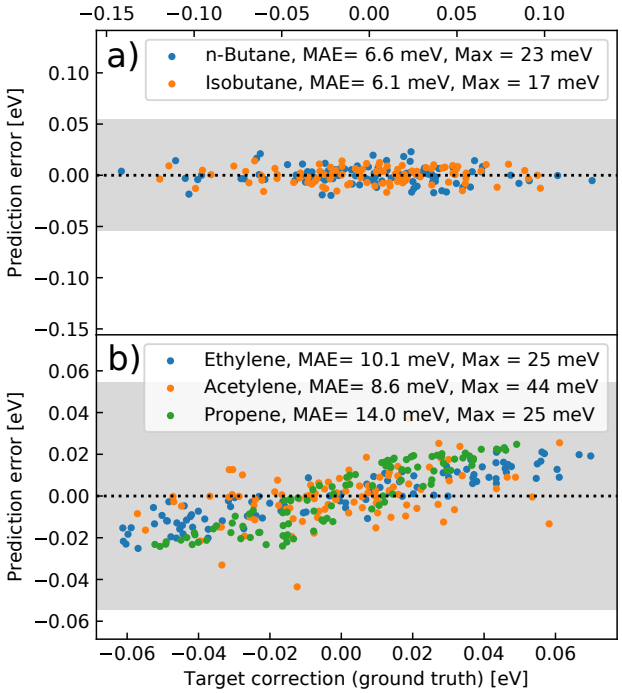

FIG. 2. Residual plots for the transferability task on small hydrocarbons. Values on the x-axis correspond to the target values given by the difference between reference (CCSD(T)) and baseline (PBE) method total energies. Prediction errors are defined as the difference between the reference total energies and the ones obtained self-consistently with NeuralXC optimized on ethane and propane.

Transferability. Beyond being data-efficient, a useful machine learning model generalizes well to unseen data. In statistical learning, it is traditionally assumed that both training and test set are i.i.d. samples of the same underlying distribution. There is no reason to believe that a model should extrapolate beyond the population on which it was trained. A neural network trained to differentiate between pictures of cats and dogs is not able to tag those containing birds, for example.

In an apparent contrast to this, we would like to create a machine-learned functional that, after being exposed to a small sample of molecules, generalizes to more complex and more extensive systems. However, even though molecules might differ significantly in their structural variables from those contained in the training set, locally, their charge distributions and, therefore, the input to the network can still be similar as long as the chemical processes involved do not change too much.

To test the transferability of our functional, we start by comparing our method to that of Cheng et al. [24]. After being trained on 50 ethane and 20 propane geometries, the model's capability of correctly reproducing relative energies for 100 n-butane and isobutane geometries

Method	Training set composition	Butane MAE	Butane Max.	Isobutane MAE	Isobutane Max.
PBE	-	37	115	36	120
MOB-ML[23]	100, 100, 50	16	60	25	87
MOB-ML(mod.)[24]	20, 50, 0	8.7	38	8.8	38
NeuralXC	20, 50, 0	**6.6**	**23**	**6.1**	**17**
NeuralXC	*5, 10, 0*	*8.6*	*31*	*9.1*	*27*

TABLE I. Mean absolute error (MAE) and maximum absolute error (Max.) for the transferability task on small alkanes. The second column describes how many samples of propane, ethane and methane were contained in the training set. Energy errors are given in meV.

is assessed. Fig. 2 shows that these energies are predicted well beyond chemical accuracy with MAEs of 6.6 meV and 6.1 meV respectively and that in fact, we are more accurate than Cheng et al.'s state of the art method which achieves MAEs of 8.7 meV and 8.8 meV. Even after the training set size was decreased to 10 ethane and 5 propane structures, our model's accuracy remains comparable to that of Cheng et al.'s, as can be seen in Tab. I.

We would further like to assess how well our model generalizes to other hybridizations of the carbon atom. Fig. 2 shows the prediction errors of the model used in Fig. 2 for the augmented test set containing systems with double and triple bonds. While we see a decline in performance for these systems, the model still predicts all total energies within 2 mHartree or 54 meV of the reference values. The positive linear correlation between prediction error and target value, which is especially prominent for ethylene and propene, suggests the existence of systematic errors. These errors are most likely due to the model's failure to treat physical effects deriving from the sp and sp^2 hybridizations of the carbon atom and could be compensated by including relevant structures in the training set.

Moving on from hydrocarbons, we created a functional that was optimized to reproduce highly accurate calculations for small water clusters. We will show that this model learns to correct hydrogen bonds involving oxygen, regardless of the chemical environment. We will further outline in the next section, how it can be used in molecular dynamics simulations of liquid water. The machine-learned functional was built as an additive correction to the PBE xc-functional and consisted of a sum of two models. The first one was trained to reproduce the total energies of monomers and dimers. The second model was then built on top of the first to correct three-body energies in trimers.

Tab. II shows the final model's generalization error compared to its baseline method on a test set consisting of 200 monomers, 500 dimers, and 250 trimers, obtained in the same way as the training set. Rather than comparing total energies, we show errors for one, two, and three-body energies as defined in Ref. [35] as otherwise large one-body energies would always dominate the comparison.

The high accuracy shown for two-body energies indicates that our optimized functional correctly treats hydrogen bonds between two water molecules. It is unclear, however, whether this extends to systems containing molecules other than water. For this purpose, we have made use of the s66x8 dataset [38] introduced above. Our findings are summarized in Fig. 3 using a representative subset of these systems: It can be observed that NeuralXC accurately treats the hydrogen bond involving an oxygen atom for a wide variety of systems. As expected, the functional fails to correct the hydrogen bond involving nitrogen, an element on which the model was not trained.

Molecular dynamics. Using our ML model as a potential instead of merely adding an energy correction as proposed in earlier work by the authors [18] and in related work [19] has the advantage that electron densities are self-consistent with respect to the underlying functional. Self-consistency makes the Hellmann-Feynman theorem [20] applicable, allowing us to obtain accurate, energy-preserving forces that can be used to study the dynamical properties of a system. We illustrate the utility of this approach to running molecular dynamics simulations for liquid water using the NeuralXC functional optimized on small water clusters.

It is commonly accepted that the accurate description of liquid water necessitates the use of hybrid functionals and the explicit treatment of dispersion forces and nuclear quantum effects (NQEs) [42]. The latter is often achieved through path integral molecular dynamics [43], the cost of which still prohibits its use in ab-initio simulations of realistically sized systems. Testing our optimized functional on liquid water, we, therefore, bear in mind that an exact agreement with experimental results could only be achieved if NQEs were to be explicitly included.

The molecular dynamics simulations was run for 96 water molecules in a periodic box at experimental density and 315 K, choosing a higher temperature to approximately accommodate for NQEs. We sampled four initial configurations, from a thermalized molecular dynamics simulation of the same system run with MB-pol, each sample being 5 ps apart. These configurations were then used together with random initial velocities as starting points for four 5 ps MD runs with time step 0.5 fs, using both PBE and NeuralXC as functionals. In the case of PBE, each initial configuration was again thermalized by

Method	1-body RMSE	MAE	Max.	2-body RMSE	MAE	Max.	Mean	3-body RMSE	MAE	Max.	Mean
PBE	61	48	174	35	21	263	6.3	10	6.2	72	-3.6
NeuralXC	1.8	1.4	9.3	9.1	6.6	42	-5.5	7.1	4.8	37	-3.8

TABLE II. Generalization errors of the model trained on water monomers, dimers and trimers. The errors in total energy are split up into their many-body contributions. For monomers the 1-body errors are reported, for dimers the 2-body errors and for trimers the 3-body errors. All values are given in meV.

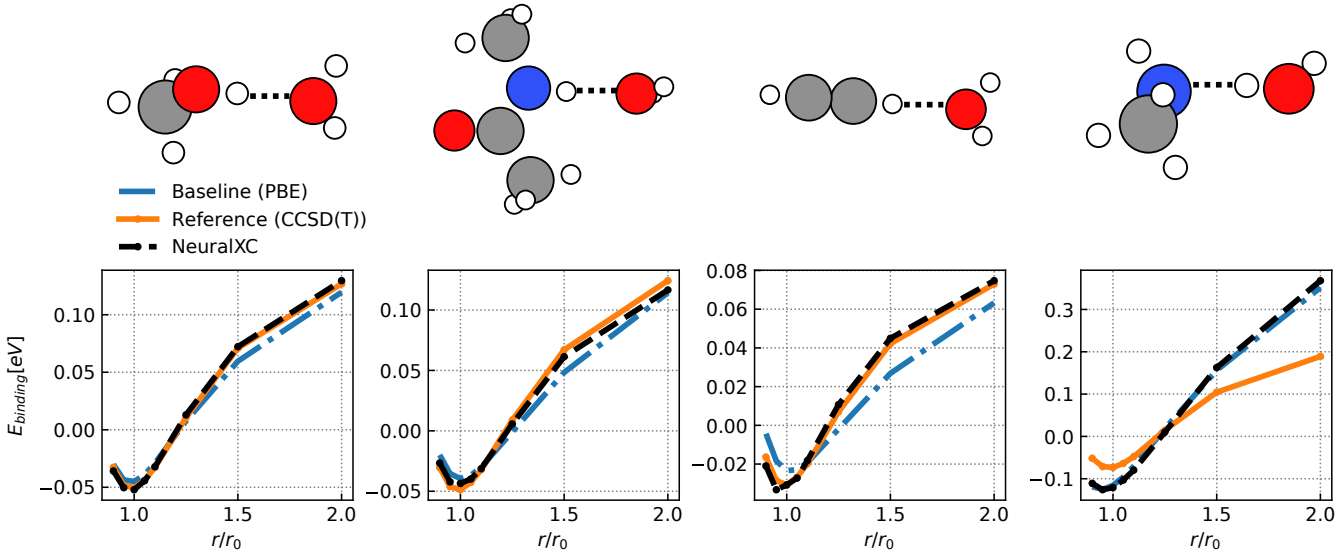

FIG. 3. Representative subset of dissociation curves taken from the s66x8 [38] dataset. From left to right, the figures describe hydrogen bonds between: a) water and methanol, b) n-methylacetamine (C_3H_7ON) and water, c) acetylene and water and d) water and methylamine (CH_3NH_2), where the molecule listed first is understood to be the hydrogen donor. The energies depicted correspond to binding energies for the reference calculations (orange) and total energies for the baseline (blue) and NeuralXC (black) calculations. As the molecules stay rigid during the relative displacement, binding and total energies merely differ by a constant. We, therefore, shifted every curve by its mean energy to align them and draw better comparisons between the methods. The depicted distances were normalized by dividing them through their respective equilibrium values, which were determined with MP2.

using a Nose-Hoover thermostat [44, 45] for 2 ps before using the velocity-verlet algorithm [46] to propagate the system.

Fig. 4a indicates that our functional is capable of accurately reproducing the oxygen-oxygen radial distribution function (RDF) obtained from x-ray diffraction experiments [40] and joint refinement of neutron and x-ray data [41]. Small deviations can be observed in the height of the first peak, which can be credited to the lack of explicit treatment of NQEs. It is understood that the height of this peak is highly sensitive to NQEs and that a more reliable measure for the quality of a functional is given by the shape of the first trough characterizing the distribution of molecules in the interstitial region [47, 48]. This minimum and subsequent maxima in the RDF are accurately reproduced by the NeuralXC functional. RDFs for OH and HH show more significant deviations from experimental results than their OO counterpart. It should be noted that due to their small mass, hydrogen atoms are more susceptible to nuclear quantum effects. We, therefore, expect these effects to play a larger role in RDFs involving hydrogen, a notion that has been confirmed by PIMD studies using the MB-pol force-field [15].

Electronic densities. So far, we have only tested properties that can be directly related to the ground state energy of a given system. The fact that we merely impose that the model reproduces total energies during training, introduces freedom as to how this energy is distributed across space (or equivalently, across orbitals). However, this distribution determines how well the model reproduces properties that go beyond ground state energies such as the (ground state) electron density and energy eigenstates of the system.

Fig. 5 compares the change in density going from PBE to CCSD(T) (left) to that of going from PBE to NeuralXC (right). Densities on the left were calculated with PySCF [34], and a cc-pVDZ basis set, whereas densities on the right, were calculated with SIESTA [26] and a numerical polarized double zeta basis set. Therefore, comparisons can only be made at a qualitative level as contributions from the choice of functional cannot be separated from basis-set effects. We can tell from the plotted

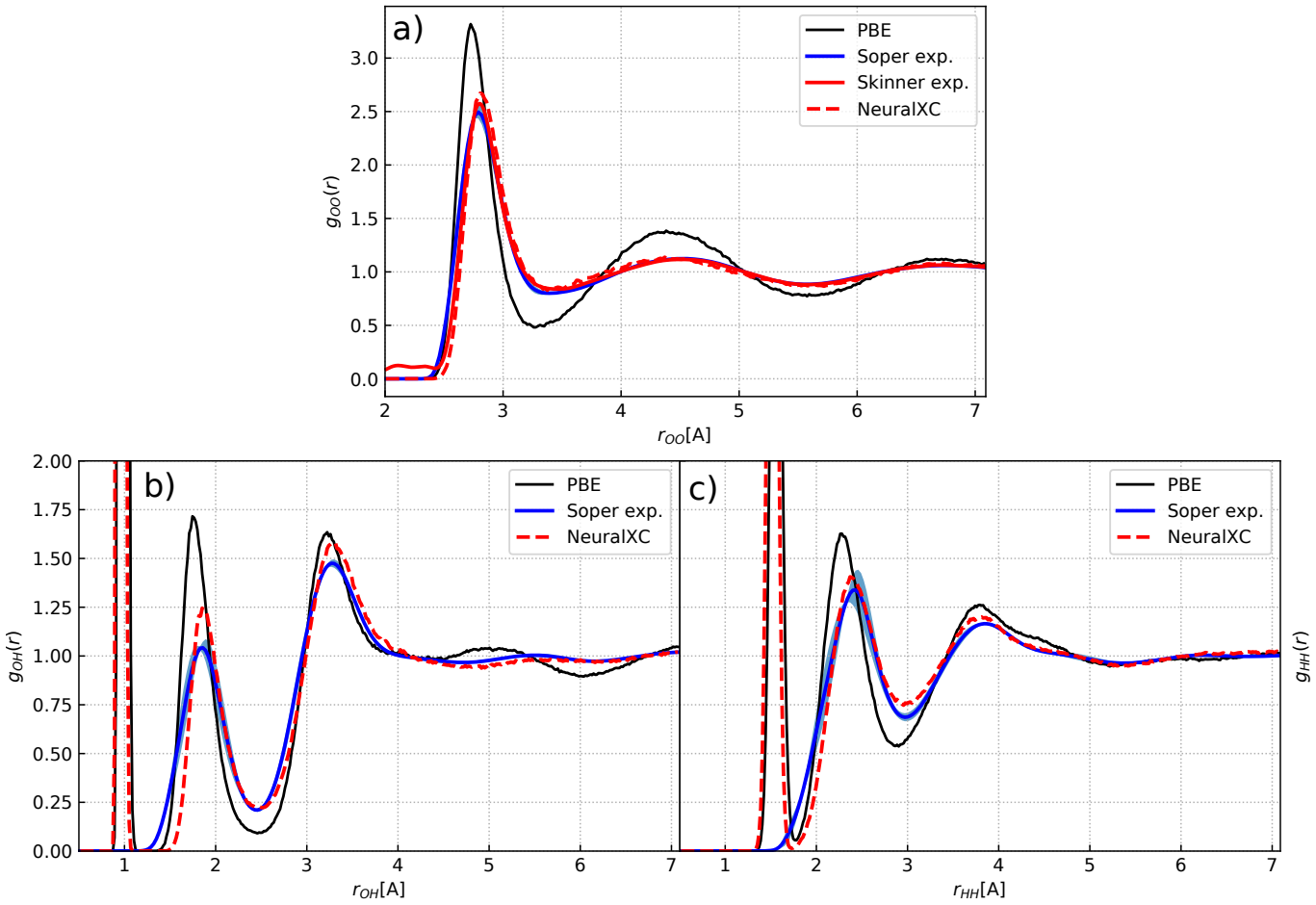

FIG. 4. Radial distribution functions (RDFs) in the order a) OO, b) OH, c) HH. The RDFs obtained from Born-Oppenheimer molecular dynamics simulations of 96 water molecules in a periodic box at experimental density and 317 K using PBE and NeuralXC as functionals are compared to experimental results by Skinner et al. [40] and Soper [41]. Uncertainties in the experimental results by Soper are shown as shaded area.

isosurfaces that there is a qualitative agreement between the two methods. Especially for benzene and toluene, both methods localize more electrons around the hydrogens and on carbon-carbon bonds, albeit CCSD(T) showing more change quantitatively. The same behavior can be observed for malonaldehyde (MDA), where the electron density is redistributed from carbon to hydrogen and oxygen atoms. In general, NeuralXC seems to fail to accurately reproduce more complicated nodal structures around the oxygen atoms, which is especially apparent in ethanol.

IV. CONCLUSION

We have developed a supervised ML method termed NeuralXC that lifts the accuracy of Kohn-Sham density functional calculations at a GGA level to that of coupled-cluster theory calculations. It was shown that the method is data-efficient in that it can reach desired accuracies with training sets containing only a few structures for simple molecules like water and benzene and tens to hundreds of structures for more challenging molecules such as ethanol and malonaldehyde. We have further demonstrated that the trained models are transferable across chemical environments: a model trained on ethane and propane predicted relative total energies of n-butane and isobutane structures with an MAE of 6.6 and 6.1 meV respectively, surpassing a state of the art method introduced by Cheng et al. [24] regarding accuracy. Moreover, a NeuralXC functional trained on water monomers, dimers, and trimers was able to accurately describe O-H hydrogen bonds between a water molecule and methanol, n-methylacetamine, and acetylene without being re-trained on these systems. Finally, we have shown that these machine-learned functionals can be used to conduct molecular dynamics simulations by reproducing radial distribution functions of liquid water close to experimental results.

We believe that NeuralXC opens up a new path to developing exchange-correlation functionals for KS-DFT calculations. As our method only introduces a linearly

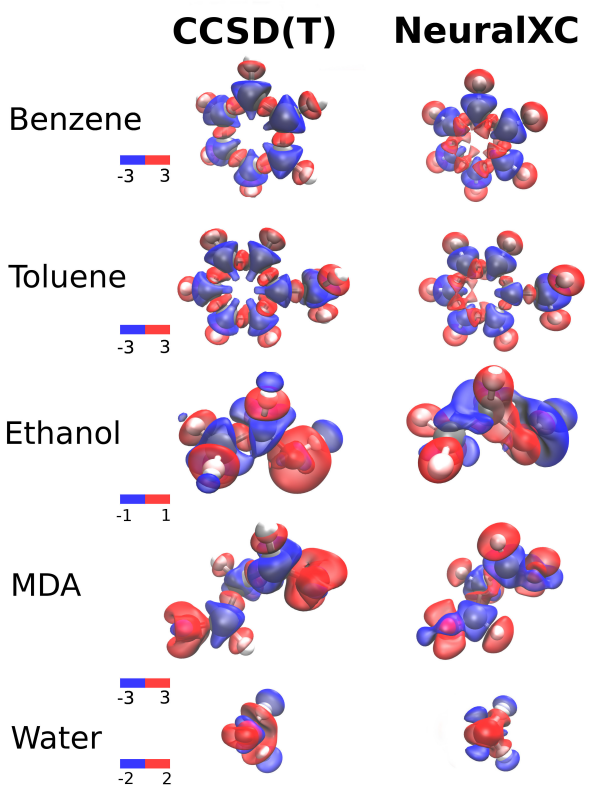

FIG. 5. Comparison of the difference in electron density between CCSD(T) and PBE (left) and NeuralXC and PBE (right). Plotted are isosurfaces of constant density change in units of 10^{-3} e/Bohr^3. The isosurface values on the left are given by the color bars depicted adjacent to the molecules, and their values were adjusted to facilitate the qualitative comparison to NeuralXC results. On the right-hand side all isosurfaces are plotted for $\pm 10^{-3}$ e/Bohr^3

scaling overhead to the underlying baseline functional, it is especially attractive for simulations of large systems for which state-of-the-art hybrid functionals are still too expensive to employ. Further research will need to address ways in which NeuralXC can correct excited state properties, to make our method suitable for more challenging tasks such as time-dependent DFT. We further see possible applications in orbital-free DFT, where NeuralXC could be used to develop kinetic energy functionals. Finally, it remains to be seen which methods can serve as suitable baselines, as e.g., the local density approximation, together with a minimal basis set, would make our method a competitive alternative to tight-binding DFT.

V. SUPPLEMENTAL DATA

Training data and ML-models will be made available upon publication. The code to train and deploy NeuralXC can be found at Ref. 49.

VI. ACKNOWLEDGEMENTS

We aknowledge funding from DOE awards numbers DE-SC0001137 and DE-SC0019394. Sebastian Dick was supported by a fellowship from The Molecular Sciences Software Institute under NSF grant ACI-1547580. We would like to thank Stony Brook Research Computing and Cyberinfrastructure, and the Institute for Advanced Computational Science at Stony Brook University for access to the high-performance SeaWulf computing system, which was made possible by a $1.4M National Science Foundation grant (#1531492). Finally, SD wants to express his thanks to Samuel Ellis for his valuable advice regarding the implementation of NeuralXC.

[1] R. O. Jones, Reviews of modern physics **87**, 897 (2015).
[2] W. Kohn and L. J. Sham, Physical review **140**, A1133 (1965).
[3] J. P. Perdew and K. Schmidt, in *AIP Conference Proceedings*, Vol. 577 (AIP, 2001) pp. 1–20.
[4] J. P. Perdew, A. Ruzsinszky, J. Tao, V. N. Staroverov, G. E. Scuseria, and G. I. Csonka, The Journal of chemical physics **123**, 062201 (2005).
[5] W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965).
[6] J. P. Perdew, K. Burke, and M. Ernzerhof, Physical review letters **77**, 3865 (1996).
[7] J. Sun, A. Ruzsinszky, and J. P. Perdew, Physical review letters **115**, 036402 (2015).
[8] Y. Zhang, D. A. Kitchaev, J. Yang, T. Chen, S. T. Dacek, R. A. Sarmiento-Pérez, M. A. Marques, H. Peng, G. Ceder, J. P. Perdew, *et al.*, npj Computational Materials **4**, 9 (2018).
[9] M. Chen, H.-Y. Ko, R. C. Remsing, M. F. C. Andrade, B. Santra, Z. Sun, A. Selloni, R. Car, M. L. Klein, J. P. Perdew, *et al.*, Proceedings of the National Academy of Sciences **114**, 10846 (2017).
[10] F. Tran, J. Doumont, L. Kalantari, A. W. Huran, M. A. Marques, and P. Blaha, Journal of Applied Physics **126**, 110902 (2019).
[11] N. Mardirossian and M. Head-Gordon, Molecular Physics **115**, 2315 (2017).
[12] N. Mardirossian and M. Head-Gordon, Physical Chemistry Chemical Physics **16**, 9904 (2014).
[13] A. D. Becke, The Journal of chemical physics **107**, 8554 (1997).
[14] M. Fritz, M. Fernández-Serra, and J. M. Soler, The Journal of chemical physics **144**, 224101 (2016).
[15] G. R. Medders, V. Babin, and F. Paesani, Journal of Chemical Theory and Computation , 140701133607002 (2014).
[16] A. Grisafi, D. M. Wilkins, B. A. Meyer, A. Fabrizio, C. Corminboeuf, and M. Ceriotti, arXiv preprint arXiv:1809.05349 (2018).
[17] A. Fabrizio, B. Meyer, M. Ceriotti, C. Corminboeuf, *et al.*, Chemical Science (2019).
[18] S. Dick and M. Fernandez-Serra, arXiv preprint arXiv:1812.06572 (2018).

[19] M. Bogojeski, L. Vogt-Maranto, M. E. Tuckerman, K.-R. Mueller, and K. Burke, .

[20] R. P. Feynman, Physical Review **56**, 340 (1939).

[21] R. Nagai, R. Akashi, and O. Sugino, arXiv preprint arXiv:1903.00238 (2019).

[22] X. Lei and A. J. Medford, Physical Review Materials **3**, 063801 (2019).

[23] M. Welborn, L. Cheng, and T. F. Miller III, Journal of chemical theory and computation **14**, 4772 (2018).

[24] L. Cheng, M. Welborn, A. S. Christensen, and T. F. Miller III, The Journal of chemical physics **150**, 131103 (2019).

[25] T. Nudejima, Y. Ikabata, J. Seino, T. Yoshikawa, and H. Nakai, The Journal of chemical physics **151**, 024104 (2019).

[26] J. M. Soler, E. Artacho, J. D. Gale, A. García, J. Junquera, P. Ordejón, and D. Sánchez-Portal, Journal of Physics: Condensed Matter **14**, 2745 (2002).

[27] J. Behler and M. Parrinello, Physical Review Letters **98**, 146401 (2007).

[28] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)* (2016) pp. 265–283.

[29] D. P. Kingma and J. Ba, arXiv preprint arXiv:1412.6980 (2014).

[30] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, in *Advances in neural information processing systems* (2007) pp. 153–160.

[31] S. Chmiela, H. E. Sauceda, K.-R. Müller, and A. Tkatchenko, , 1 (2018), arXiv:1802.09238.

[32] L. Cheng, M. Welborn, A. S. Christensen, and T. F. Miller, "Thermalized (350k) qm7b, gdb-13, water, and short alkane quantum chemistry dataset including mob-ml features," (2019).

[33] T. H. Dunning Jr, The Journal of chemical physics **90**, 1007 (1989).

[34] Q. Sun, T. C. Berkelbach, N. S. Blunt, G. H. Booth, S. Guo, Z. Li, J. Liu, J. D. McClain, E. R. Sayfutyarova, S. Sharma, S. Wouters, and G. K. Chan, "Pyscf: the pythonbased simulations of chemistry framework," (2017), https://onlinelibrary.wiley.com/doi/pdf/10.1002/wcms.1340.

[35] V. Babin, C. Leforestier, and F. Paesani, Journal of Chemical Theory and Computation **9**, 5395 (2013).

[36] V. Babin, G. R. Medders, and F. Paesani, Journal of Chemical Theory and Computation **10**, 1599 (2014), arXiv:arXiv:1210.7022v1.

[37] A. P. Bartók, R. Kondor, and G. Csányi, Physical Review B **87**, 184115 (2013).

[38] J. Rezác, K. E. Riley, and P. Hobza, Journal of chemical theory and computation **7**, 2427 (2011).

[39] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller, Science advances **3**, e1603015 (2017).

[40] L. B. Skinner, C. Huang, D. Schlesinger, L. G. Pettersson, A. Nilsson, and C. J. Benmore, The Journal of chemical physics **138**, 074506 (2013).

[41] A. K. Soper, ISRN Physical Chemistry **2013** (2013).

[42] R. A. DiStasio Jr, B. Santra, Z. Li, X. Wu, and R. Car, The Journal of chemical physics **141**, 084502 (2014).

[43] D. Marx and M. Parrinello, The Journal of chemical physics **104**, 4077 (1996).

[44] S. Nosé, The Journal of chemical physics **81**, 511 (1984).

[45] W. G. Hoover, Physical review A **31**, 1695 (1985).

[46] W. C. Swope, H. C. Andersen, P. H. Berens, and K. R. Wilson, The Journal of Chemical Physics **76**, 637 (1982).

[47] A. P. Bartók, M. J. Gillan, F. R. Manby, and G. Csányi, Physical Review B **88**, 054104 (2013).

[48] J. Wang, G. Román-Pérez, J. M. Soler, E. Artacho, and M.-V. Fernández-Serra, The Journal of chemical physics **134**, 024516 (2011).

[49] `https://github.com/semodi/neuralxc`.