The Journal of Chemical Physics

RESEARCH ARTICLE | JANUARY 05 2024

Development of a machine learning finite-range nonlocal density functional

Special Collection: John Perdew Festschrift

Zehua Chen ; Weitao Yang

J. Chem. Phys. 160, 014105 (2024)
https://doi.org/10.1063/5.0179149

The Journal of Chemical Physics ARTICLE pubs.aip.org/aip/jcp

Development of a machine learning finite-range nonlocal density functional

Cite as: J. Chem. Phys. 160, 014105 (2024); doi: 10.1063/5.0179149
Submitted: 29 September 2023 • Accepted: 12 December 2023 •
Published Online: 5 January 2024

Zehua Chen[1] and Weitao Yang[2,a)]

AFFILIATIONS

[1] Department of Chemistry, Duke University, Durham, North Carolina 27708, USA
[2] Department of Chemistry and Department of Physics, Duke University, Durham, North Carolina 27708, USA

Note: This paper is part of the JCP Festschrift for John Perdew.
[a)]**Author to whom correspondence should be addressed:** weitao.yang@duke.edu

ABSTRACT

Kohn–Sham density functional theory has been the most popular method in electronic structure calculations. To fulfill the increasing accuracy requirements, new approximate functionals are needed to address key issues in existing approximations. It is well known that nonlocal components are crucial. Current nonlocal functionals mostly require orbital dependence such as in Hartree–Fock exchange and many-body perturbation correlation energy, which, however, leads to higher computational costs. Deviating from this pathway, we describe functional nonlocality in a new approach. By partitioning the total density to atom-centered local densities, a many-body expansion is proposed. This many-body expansion can be truncated at one-body contributions, if a base functional is used and an energy correction is approximated. The contribution from each atom-centered local density is a single finite-range nonlocal functional that is universal for all atoms. We then use machine learning to develop this universal atom-centered functional. Parameters in this functional are determined by fitting to data that are produced by high-level theories. Extensive tests on several different test sets, which include reaction energies, reaction barrier heights, and non-covalent interaction energies, show that the new functional, with only the density as the basic variable, can produce results comparable to the best-performing double-hybrid functionals, (for example, for the thermochemistry test set selected from the GMTKN55 database, BLYP based machine learning functional gives a weighted total mean absolute deviations of 3.33 kcal/mol, while DSD-BLYP-D3(BJ) gives 3.28 kcal/mol) with a lower computational cost. This opens a new pathway to nonlocal functional development and applications.

I. INTRODUCTION

Among many electronic structure methods, because of its favorable accuracy and reasonable cost, density functional theory (DFT)[1,2] is most often used in very broad applications, ranging from predicting molecular and bulk material structures, studying chemical reaction mechanisms, to understanding charge and energy transfers. Although the DFT is formally exact, its success is attributed to the Kohn–Sham scheme[2] and approximations to the exchange–correlation energy. Many semi-local density functionals approximations (DFAs) have been developed.[3–8] Great success has been achieved by semi-local approximations, while challenges still present. Semi-local DFAs are known to have large delocalization errors.[9,10] Commonly used semi-local DFAs also fail to describe the van der Waals interaction.[11] Even in applications where the above issues are considered insignificant, semi-local DFAs may still produce results with insufficient accuracy. Systematic improvements of (meta-)generalized gradient approximations (GGAs) with data indicate that, the local form might limit the accuracy that semi-local functionals can achieve.[12] Nonlocal components in the functional are necessary to overcome these difficulties.

Nonlocal functionals have already been developed and widely used for many years. One way to introduce nonlocality into the functional is to mix Hartree–Fock nonlocal exchange into a semi-local functional, making the resulting approximation a hybrid functional.[13,14] Hybrid functionals partially reduce the delocalization error[10] and are generally more accurate than semi-local functionals. In recent years, double-hybrid functionals have attracted much research interest.[15–18] In addition to the nonlocal exchange, double-hybrid functionals use nonlocal correlation energies, which are usually derived from many-body perturbation theories, such as the second-order Møller–Plesset perturbation theory (MP2) and

the random phase approximation (RPA). Double-hybrid functionals outperform approximations in other categories, but at higher computational costs.

Though hybrid and double-hybrid functionals have appealing accuracy, they still have significant delocalization and/or static correlation error for general applications across systems and sizes.[19] In addition, the relatively high computational cost may prohibit their practical applications, especially when used to calculate large molecules and periodic systems. It remains a challenge on how to go beyond hybrid and double-hybrid functionals. Is there a way to design a nonlocal DFA such that orbitals are not required and its performance is possible to be systematically improved?

The main challenge is that, the nonlocal exchange–correlation energy is much harder to approximate. In some instances, the nonlocal contributions can be explicitly derived. All approximate treatments of the van der Waals interactions involve nonlocal contributions, at different levels of approximations, from empirical corrections, the use of exchange-hole, to approximations to the random phase approximations.[20–29] There is no known general pathway to build nonlocal contributions.

To explore developing DFAs, we focus on machine learning (ML), which has generated major impact in science and technology. For example, ML has been used in predicting protein structures,[30,31] constructing potential energy surfaces based on quantum mechanical calculations,[32–34] calculating total energies of molecules,[35–37] and developing general molecular force fields.[38–42] Recent reviews have described some updated progress.[42–47]

ML has also generated much interest in DFT. The application of artificial neural networks was made very early on for approximating the exchange–correlation potentials from electron densities[48] and for improving exchange–correlation density functional approximations based on a given DFA.[49] In the modern era of machine learning, many different ML approaches have been applied in various aspects of DFT, approximating the Kohn–Sham kinetic energy functional,[50–52] improving bandgap prediction,[53–55] learning the functional derivative discontinuity,[56] bypassing solving the Kohn–Sham equations,[57–62] approximating the Kohn–Sham Hamiltonian,[63,64] learning a DFA,[65] directly approximating the Kohn–Sham exchange–correlation potentials,[66–68] learning the electron density,[69] running direct molecular dynamics,[70] allowing the direct use of nonlocal pseudopotentials in orbital-free DFT,[71] describing excited states[72] and core electron binding energies,[73] and finally improving the exchange–correlation energy functional,[74–81] a key challenge in DFT. Progress in ML for DFT has been summarized in some recent reviews.[47,82–85] There are two main challenges in developing ML for the exchange–correlation energy functional.

First, to make a ML approximation to the universal exchange–correlation density functional, the invariance of the resulting functionals with respect to translation and rotation of the molecular geometries and permutation of identical atoms are necessary. The same requirement for ML has been addressed in learning the total molecular energy.[32–34] The satisfaction of the symmetry requirement can be achieved readily when the energy or energy density is expressed directly as a function of electron density and the Kohn–Sham density matrix, as in many ML developments, including the DM21.[76] A more general approach from real space distribution information is possible: Behler and Parrinello represented the total energy as a sum of atomic contributions and used a set of atom-center basis functions to construct input features.[32,86] This has been used in may ML applications,[87] including approximating the exchange–correlation energy functional[78] and learning the bandgaps of materials.[55] This framework has also been used in our work on accelerating free energy calculations in hybrid QM/MM (quantum mechanics/molecular mechanics) simulations[88,89] and developing general force fields.[90–92] An interesting recent development in preserving the rotation symmetry is the harmonic networks in convolution neural networks;[93,94] its application in learning a DFA shows promise.[79]

Second, a particularly challenging aspect in approximating the exchange–correlation energy functional is the nonlocal quantum effects. While the classical Coulomb interaction is nonlocal, its functional is known and is used in all DFT calculations with any DFA. The remaining nonlocal effect is in the quantum mechanical exchange–correlation energy: Electron density far away from a given point in space can play an important role in the exchange–correlation energy contribution from this point. In other words, the exchange–correlation contributions from electron density far apart can be correlated, not simply as additive. Key exact constraints of the density functionals demand such nonlocal contributions: the fractional charge conditions presented in the work of Perdew *et al.*,[95] the fractional spin conditions presented in the work of Cohen *et al.*,[96,97] and the flat-plane combined conditions of fractional charges and spins presented in the work of Mori-Sanchez *et al.*[98] The violation of these exact conditions in the commonly used DFAs, which do not have such nonlocal contributions, leads to systematic delocalization and static correlation errors.[9,99,100] Therefore, the nonlocal contributions needed for satisfying these exact conditions are critical for DFT development.

Another prominent nonlocal effect is the van der Waals interaction.[22,23,25,26,101,102] Not only is it nonlocal, it is also a many-body correlation.[29] The nonlocal van der Waals interaction is critical for intermolecular interactions in many chemical, biological, and materials systems.

In the present work, we explore the development of a finite-range nonlocal functional based on a many-body expansion and ML. Because of the limitation of finite range, it only includes partial nonlocal contributions. We will describe our idea on how such a development can be compatible and combined with recent analytical developments in nonlocal functional contributions, which can be ultralong-range.[103,104]

The rest of this paper is organized as follows: In Sec. II, we present the theoretical justification for using the atom-center expansion for energy correction, as well as details about the procedure of input generation for machine learning. Section III presents computational details. Section IV provides an analysis of the performance of the machine learning functional with discussions on how to further improve it with more training data. Finally, we summarize our work in Sec. V.

II. THEORY

A. Many-body expansion in DFT

The many-body expansion is commonly used in computational chemistry to reduce the computational cost. Its success has been witnessed in fragment based methods[105–107] and the method of increments.[108,109] In terms of the total electron density, a similar

definition for the many-body expansion can also be made: A part of the total energy is expressed as a summation of one-body and two-body energies and terms that consist of more bodies,

$$\Delta E[\rho] = \sum_i \Delta E_1[\rho_i] + \sum_{i<j} \Delta E_2[\rho_i, \rho_j] + \sum_{i<j<k} \Delta E_3[\rho_i, \rho_j, \rho_k] + \cdots, \tag{1}$$

where ρ is the total electron density and ρ_i is the density for one body. The choice of the one-body density ρ_i is crucial for the quality of the many-body expansion. One-body densities should be local, such that the many-body expansion can converge. It is natural to define the body as the local density around the nuclei. In this work, the Hirshfeld partition scheme[110] is used to generate the local densities:

$$\rho_i(\mathbf{r}) = \frac{\rho_i^0(|\mathbf{r} - \mathbf{R}_i|)}{\sum_a \rho_a^0(|\mathbf{r} - \mathbf{R}_a|)} \rho(\mathbf{r}), \tag{2}$$

where ρ_a^0 is the spherical electron density of free atom a, although this is only one of many possible choices. See Fig. S1 for an example on how the partitioned density compares to the total density.

It is also important to determine which part of the total energy should be approximated by the many-body expansion. The total energy, or even the much smaller exchange–correlation energy, may require many terms in the many-body expansion to achieve good accuracy. We demonstrate this point here. As shown in Fig. 1, the one-body expansion approximation fails to recover the BLYP[5,6,114] exchange–correlation energy. A much smaller quantity, the difference between total energies produced by different DFAs, can be much better approximated. This approximating difference approach

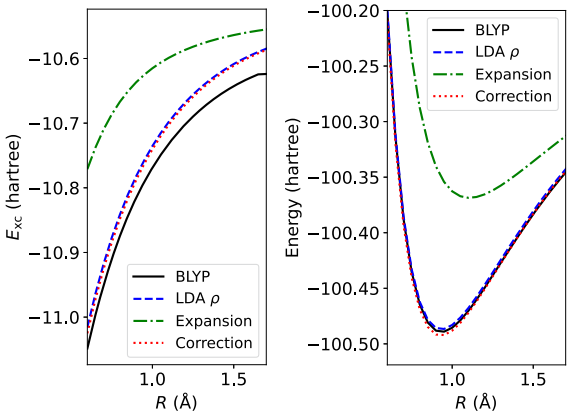

FIG. 1. Left: Exchange–correlation energy in HF for nuclear separations from 0.6 to 1.7 Å. The "BLYP" curve means E_{xc}^{BLYP} evaluated with BLYP self-consistent density (ρ^{BLYP}). The "LDA ρ" curve is E_{xc}^{BLYP} evaluated with LDA self-consistent density (ρ^{LDA}). The curve labeled as "Expansion" means we directly decompose E_{xc} using partitioned densities: $E_{xc} = E_{xc}^{BLYP}[\rho_H^{LDA}] + E_{xc}^{BLYP}[\rho_F^{LDA}]$. The "Correction" curve is computed by using $E_{xc} = E_{xc}^{LDA}[\rho^{LDA}] + (E_{xc}^{BLYP} - E_{xc}^{LDA})[\rho_H^{LDA}] + (E_{xc}^{BLYP} - E_{xc}^{LDA})[\rho_F^{LDA}]$. Right: Total energy of HF molecule computed in the ways as discussed before. It is noteworthy that even though there is a difference in the exchange–correlation energy when using different densities, the total energy still agrees very well. This can be attributed to the variational property of the total energy.

was already used in the early work on machine learning for DFT in the work of Zheng *et al.*[49] and is now known as the Δ learning.[46] Therefore, to truncate at the one-body terms, we express the DFT total energy as

$$E[\rho] \approx E^{base}[\rho] + \sum_i \Delta E_1[\rho_i], \tag{3}$$

where E is the exact energy and E^{base} is the ground level we start from. As pointed out before, E^{base} should already contain an approximate E_{xc} (usually a semi-local functional), otherwise terminating at one-body terms will not suffice. Even if in Fig. 1, we have shown that, with the aid of an one-body correction, E^{BLYP} can be recovered from a local density approximation (LDA) calculation; this observation, however, does not necessarily apply to the exact energy. Long-range correlation effects, such as the van der Waals interaction, may require two and more body terms, if the base functional does not fully capture the long-range correlation. In this work, we capture the two-body van der Waals interaction with the DFT-D3 model,[112,113] similar to the development of DM21.[76] Note that other many-body interactions, such as many-body van der Waals interactions,[29] and ultralong-range effects, corrections to delocalization and static correlation errors, are not considered presently.

In Eq. (3), we retain only the one-body terms in the many-body expansion of Eq. (1), as throughout present work. This form with one-body terms only is then the atom-center expansion in the work of Behler and Parrinello for neural network learning.[32] In machine learning for exchange and correlation energy in DFT, this atom-center expansion was also used in NeuralXC[78] and DeePKS.[80]

The usage of one-body correction is based on the observation that the effects are mostly "local." However, it does not imply that the resulting functional is local. Indeed, $\Delta E_1[\rho_i]$ is relatively "local" as compared to the entire chemical system. Its functional form, however, can be completely nonlocal in terms of the atom-centered density ρ_i. Hence, each one-body term $\Delta E_1[\rho_i]$ may be called a finite-range nonlocal functional. The finite-range character comes from the fact that ρ_i resides only within a certain range. Beyond that range, ΔE_1 does not play a role. This greatly simplifies the procedure of approximating ΔE_1. By expressing the approximate functional as a base semi-local functional plus one-body corrections, although the resulting functional is not completely nonlocal, it already goes beyond traditional semi-local functionals. The combination of locality in the atom scale and nonlocality in terms of the functional form enables the development of a density functional that captures most of the short-range correlation effects, without needing virtual orbitals.

One feature in our work, as expressed in Eq. (3), is that we only have a single functional, $\Delta E_1[\rho]$, for any system, molecules, or bulk materials. This design has been used in a previous work,[80] but many previous atom-center expansions use atom-dependent expressions for the one-body expansion.[32,78] The use of atom-independent one-body expansion is possible because the one-body electron densities, as in Eq. (2), encode the information of electron density and hence the atom. Thus, only one universal functional $\Delta E_1[\rho_i]$ is needed. Likewise, in future development when we include two-body terms in the expansion of Eq. (1), we only need another

universal functional $\Delta E_2[\rho_i, \rho_j]$ of two-body densities. This greatly simplifies the model development and also its applications. After reducing to atom-centered subproblems, the energy correction automatically becomes translationally and permutationally invariant and size-extensive. This removes the restrictions that one may encounter when designing a global nonlocal functional, thus making it possible to machine-learn the exchange–correlation energy functional (by learning the one-body energy correction).

B. Machine learning the finite-range nonlocal functional

Even if the subproblems seem simple, an explicit functional form is still impractical to be derived for $\Delta E_1[\rho_i]$. Machine learning is an excellent tool to develop this finite-range nonlocal functional numerically. Machine learning has already been applied in quantum chemistry,[114,115] and particularly, in the field of DFT.[47–57,63–79,82–85,116] In this work, the subproblem is suitable to be solved by machine learning, because of the finite-range character of the local density ρ_i. ρ_i is an atom-centered density that is partitioned from the molecular total density, and in general it is not too different from the electron density of free atom i. Therefore, by feeding high-quality data for various element types to the machine learning model, it is possible that this one-body energy correction can be predicted for any given input local density (in the interpolation range), with a low computational cost. Then, the correction to the total energy of the chemical system is a simple summation of atomic contributions, as given by the many-body expansion.

Designing an approximate functional by machine learning is very distinct from what has been done for conventional functionals. Currently available functionals all strive for as few parameters as possible and are often fitted to exact constraints and limited number of data. In this way, the functional developed hopefully can be applicable in most systems. By contrast, machine learning methods usually have a huge amount of parameters, rely heavily on the training data, and are expected to produce good interpolation quality. In this sense, it may appear that machine learning is not relevant to functional development; however, because the nonlocal density functional has no known explicit form, machine learning is in fact particularly suitable for this task because of its black box nature. As long as enough training data are provided, there is no prior knowledge required on the form of the function fitted. The use of data in the functional development has already been well established in the development of functionals like Minnesota functionals[117,118] and ω-B97 series.[119,120] Machine learning is simply another tool to utilize the data.

To express the finite-range nonlocal functional as a machine learning function, the local density needs to be preprocessed to provide finite number of features as inputs. A conventional DFA, such as a generalized gradient approximation (GGA), needs to be provided as the base functional $E_{\mathrm{xc}}^{\mathrm{base}}$, and the base total energy functional is expressed as

$$E^{\mathrm{base}}[\rho] = \sum_{A<B} \frac{Z_A Z_B}{|\mathbf{R}_A - \mathbf{R}_B|} - \int \rho(\mathbf{r}) \sum_A \frac{Z_A}{|\mathbf{R}_A - \mathbf{r}|}\, \mathrm{d}\mathbf{r} + \frac{1}{2} \iint \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|}\, \mathrm{d}\mathbf{r}\, \mathrm{d}\mathbf{r}' + T_{\mathrm{s}}[\rho(\mathbf{r})] + E_{\mathrm{xc}}^{\mathrm{base}}[\rho(\mathbf{r})]. \quad (4)$$

The base exchange–correlation functional is used not only to provide the base energy but also to produce the density.

The goal of this work is to use a machine learning functional (f) to approximate the unknown one-body energy correction functional (ΔE_1) in Eq. (3). To set the target for machine learning, we require that even with only one-body terms and the DFT-D3 dispersion energy[112,113] (with the Becke and Johnson damping[23]), accurate energies generated by high-level theories can be precisely reproduced:

$$\sum_i f[\rho_i(\mathbf{r})] + E_{\mathrm{D3}} \equiv E^{\mathrm{accurate}}[\rho(\mathbf{r})] - E^{\mathrm{base}}[\rho(\mathbf{r})]. \quad (5)$$

Here, the D3(BJ) dispersion energy is to partially recover the ignored two-body terms in the many-body expansion and the parameters for the D3(BJ) model are to be re-optimized during the training process of the machine learning model. Many machine learning models are available and each serves its particular purpose. In this work, we use the feed-forward artificial neural network (multilayer perceptrons, MLP), and the theory developed here should apply in most other machine learning models. Because neural networks do not have any built-in symmetry, special care needs to be taken when designing the input. The many-body expansion already takes care of the translational and permutational invariance; hence, the rotational invariance should be guaranteed by the transformed input. One common choice of making rotational invariant features is using the power spectrum:[121]

$$p_{nl}^{(A)} = \sum_{m=-l}^{l} (c_{nlm}^{(A)})^* c_{nlm}^{(A)}. \quad (6)$$

Here, $c_{nlm}^{(A)}$ is the coefficient that the density projects on a basis function, which is centered at atom A,

$$c_{nlm}^{(A)} = \langle \rho_A(\mathbf{r}) | g_n(|\mathbf{r} - \mathbf{R}_A|) Y_l^m(\widehat{\mathbf{r} - \mathbf{R}_A}) \rangle, \quad (7)$$

Y_l^m's are spherical harmonics, and g_n is a Gaussian function given by

$$g_n(r) = \left(\frac{2\sigma_n}{\pi}\right)^{\frac{1}{4}} \exp\left[-\sigma_n (r - \mu_n)^2\right], \quad (8)$$

where $\{\mu_n\}$ is the set of nodes that the radial functions are sampled on and σ_n is determined in the following way:

$$\sigma_n = \begin{cases} \dfrac{c^2}{(\mu_1 - \mu_0)^2}, & n = 0, \\ \dfrac{4c^2}{(\mu_{n+1} - \mu_{n-1})^2}, & 0 < n < n_{\max} - 1, \\ \dfrac{c^2}{(\mu_n - \mu_{n-1})^2}, & n = n_{\max} - 1. \end{cases} \quad (9)$$

In this work, $\{\mu_n\}$ is chosen to be the set that includes zero and scaled roots of the Gauss–Laguerre polynomial (linearly scaled to make the largest root become $\mu_{\max}$, a given parameter), and c is another adjustable parameter. Figure S2 shows a typical radial function setup used in this work. Since the creation of the power spectrum for each atom is limited to a finite region around the atom, this further imposes a limit on the range that the nonlocal functional can explore.

With the power spectrum to discretize the local density, the machine learning functional in Eq. (5) turns to an explicit function of the power spectrum:

$$f[\rho_i(\mathbf{r})] = f(\{p_{nl}^{(i)}\}). \tag{10}$$

It should be highlighted that this function is a universal one. In other words, it is not tied to a specific molecule or element type and will be used for all entries in the database.

To handle spin-unrestricted calculations properly, we need to keep the symmetry that $E[\rho^{\uparrow},\rho^{\downarrow}] = E[\rho^{\downarrow},\rho^{\uparrow}]$ (when no external magnetic field is applied). The goal can be easily achieved by using the summation of spin up and down power spectra as the input for the neural network:

$$\begin{aligned}\Delta E[\rho^{\uparrow},\rho^{\downarrow}] &= \sum_i f(\{p_{nl}^{(i)\uparrow} + p_{nl}^{(i)\downarrow}\}) = \sum_i f(\{p_{nl}^{(i)\downarrow} + p_{nl}^{(i)\uparrow}\}) \\ &= \Delta E[\rho^{\downarrow},\rho^{\uparrow}].\end{aligned} \tag{11}$$

Notice that this is not the same as taking the total density $\rho^{\uparrow} + \rho^{\downarrow}$ as the input; hence, it is possible that this form can take into account the effect of spin polarization.

The details about the neural network and the form of the cost function used in this work are described in Sec. III. Here, we describe the way of scaling the data when constructing the cost function. Since reaction energies in the database can differ in the order of magnitude, the reaction energy data need to be scaled first to avoid ignoring small numbers. We follow the idea of weighted total mean absolute deviations (WTMADs) as presented in Ref. 122. The reaction energy and stoichiometry for each reaction are scaled by

$$\text{scale} = \frac{81.24\ \text{kcal/mol}}{\overline{|\Delta E_i|}}, \tag{12}$$

where $\overline{|\Delta E_i|}$ is the mean reaction energy for subset i that the reaction belongs to. 81.24 kcal/mol is the average value of $\overline{|\Delta E_i|}$ for all subsets.

III. COMPUTATIONAL DETAILS

The accuracy of the machine learning model is mostly controlled by the quantity and quality of the training data. The data we used in this work are from MGCDB84[19] and GMTKN55[122] databases. These databases include non-covalent interaction energies, isomerization energies, thermochemistry, and barrier heights, whose values are compiled from many previous works that stand as best theoretical estimates. The methods used are usually coupled cluster with singles, doubles, and perturbative triples [CCSD(T)][123,124] at the complete basis set (CBS) limit, and composite methods such as Weizmann-4 theory.[125–127] Some modifications were done to remove duplicated subsets.

To train and test and neural network, we mainly used reaction energy data that have elements in first four rows of the periodic table. In Fig. S3, we show how many times each element appears in the molecules that are used in the database. We reserve 10% of the total data as the test set, while the remaining 90% is used for training. The GMTKN55 database has some data involving main group heavy elements, which will first be completely excluded from the training set, and later they will be used to show the transferability of the machine learning functional upon the addition of new training data.

The AE18 set (absolute atomic energies of hydrogen through argon)[128] from MGCDB84 is used purely for training. In addition, it does not participate in the dataset splitting process, neither does it receive the WTMAD2 scaling as described in Eq. (12) (instead, the scale is 10). In other words, every training task uses the entire atomic absolute energy set regardlessly, and the set gets a constant amplification of 10. This is because they are precious appropriate norms that control the absolute energies the machine learning functional is going to produce.

In this work, BLYP[5,6,114] is chosen as the base GGA functional. Gaussian 16[129] is used for base functional calculations and the resulting wave functions are stored as extended wave function files (.wfx files). Def2-QZVPPD basis[130] is used. A pruned (99, 590) grid and tight convergence criteria (root of mean squared change in the density matrix $<10^{-8}$) are used for most of the calculations, with the exception that a finer grid (500, 974) is adopted in the calculations of 18 atom absolute energies. Broken-symmetry unrestricted calculation is preferred if it lowers the total energy. For calculations that involve elements that are heavier than Kr, empirical core potentials (ECPs) are used. However, core electron densities from ECPs are not used in the following computation of the power spectrum.

The generation of the power spectra from densities is done by a Python script utilizing PySCF,[131,132] which can be accessed according to the Data Availability Statement. It uses the wave function file generated by Gaussian 16 to reproduce the density and computes $c_{nlm}^{(i)}$ numerically with standard DFT integration grids. It also supports the generation of power spectra from a PySCF DFT calculation. Free atom calculations are also performed by Gaussian 16 with high-spin configurations. These free atom densities are averaged over the angular degree of freedom to produce spherically symmetric free atom densities, which will be used in the Hirshfeld density partition.

The number of power spectra elements reflects how accurately the density is discretized. It is determined by the number of radial functions ($n_{\max}$) and the angular momentum cutoff in the angular expansion ($l_{\max}$). Usually the more radial functions are used, and the higher angular momentum cutoff is set, the better quality the discretization will have. However, a high resolution in the power spectra does not necessarily improve the performance of the neural network. In this work, we will experiment with $n_{\max}$ from 5 to 15, and $l_{\max}$ from 2 to 6.

For each molecule, the terms in the DFT-D3(BJ) energy expression are computed and output to data files with a locally modified `dftd3` program.[133] From these data files, during the neural network training process, the training script can conveniently produce DFT-D3(BJ) energies while adjusting DFT-D3(BJ) parameters (s_6, s_8, a_1, and a_2).

The machine learning function in this work is to estimate the atomic energy correction from the power spectrum generated by the local partitioned density, as explained by Eqs. (5) and (10). Therefore, the size of the input layer is tied to the shape of the power spectrum, as specified by the parameters $n_{\max}$ and $l_{\max}$. The output layer has a single node to directly produce the energy correction. The neural network model used in this work is fixed to have two hidden layers (we will experiment with 16, 32, and 64 neurons per layer in

this work), thus this model is controlled by three weight matrices ($\mathbf{W}^0$, $\mathbf{W}^1$, and $\mathbf{W}^2$) and three bias vectors ($\vec{b}^0$, $\vec{b}^1$, and $\vec{b}^2$). Hyperbolic tangent is used as the activation function for hidden neurons. The cost function used to optimize the neural network is defined as

$$L(\{\mathbf{W}^k\},\{\vec{b}^k\}) = \frac{1}{N}\sum_i^N \log\left(\cosh\left(\hat{y}_i - y_i\right)\right) + \beta\sum|w_{ij}^k|, \quad (13)$$

where N is the number of entries in the database, $\hat{y}$ is the reference value, and y is the value produced by the base GGA along with the neural network energy correction, given the set of weights $\{\mathbf{W}^k\}$ and biases $\{\vec{b}^k\}$. The second term in the cost function is the L1 regularization, which is used to prevent overfitting. Since a single neural network function f produces an energy correction for one atom in a molecule, while reference data are usually reaction energies, in most cases y is a linear combination of atomic contributions. For example, for reaction X + Y → Z, the reaction energy produced by this functional is

$$y = E_{\mathrm{Z}}^{\mathrm{base+D3}} - E_{\mathrm{X}}^{\mathrm{base+D3}} - E_{\mathrm{Y}}^{\mathrm{base+D3}} + \sum_{i\in\mathrm{Z}} f(\{p_{nl}^{(i)}\}) - \sum_{j\in\mathrm{X}} f(\{p_{nl}^{(j)}\}) - \sum_{k\in\mathrm{Y}} f(\{p_{nl}^{(k)}\}). \quad (14)$$

PyTorch[134] is used to train the model and the Adam optimizer[135] is used (with learning rate = 0.001). The weights of the neural network are initialized with the Kaiming uniform initializer[136] and the biases are zero-initialized. 10-fold cross-validation (CV) is used to produce cross-validation errors when tuning the parameters in the input features and hyperparameters in the training process. For each split, one fold is used to produce the test score, and the remaining nine folds are divided as eight folds for training and one fold for validation (to early-stop the training process). Optimal hyperparameters are determined as follows: 512 for the batch size and 5×10^{-8} for the L1 regularization factor.

We note that in this work, the resulting neural network functional is only used as a post-SCF energy correction. It is possible to achieve self-consistency, as through the iterative training procedure described in Ref. 78; however, iterative training is computationally demanding when such a large dataset and large basis sets are used. Though there exist regularized training methods that can avoid iterative training,[137] high-quality reference densities are required as inputs. Unfortunately, MGCDB84 and GMTKN55 are purely energy databases with no density/force labels provided. Without high-quality densities/forces to regularize the training, it is possible that the resulting densities/forces can be inaccurate or even unphysical, and when worse densities are fed to the model, worse energy predictions can be produced. In future developments, a self-consistent version of this model could be developed using base functional densities as the regularizer, and high-quality density/force data can be amended to the training set used here.

The computational cost of our method involves two parts: the cost of creating the power spectra and the cost of neural network computations. The cost of evaluating a trained neural network is negligible, and there are N neural networks to compute for a molecule with N atoms. The more expensive part is the generation of power spectra, which involves numerical integrations. However, it is still much cheaper than the base functional calculation ($\mathcal{O}(N^3)$), because it at most needs $\mathcal{O}(N^2)$ calculations (there are N atoms, and for each atom, the program needs to loop over the numerical grid of the entire molecule). For large molecules, the cost can be reduced to close to $\mathcal{O}(N)$, since Gaussian radial functions decay very quickly beyond a certain range and we set a cutoff radius for the numerical integration.

IV. RESULTS AND DISCUSSION

A. Optimal power spectra parameters

Although a larger neural network usually has a higher capability of learning the unknown function, using an oversized neural network along with limited number of training data may risk overfitting. In this work, there are around 5000 training data points, while the number of parameters in the neural network can easily exceed this quantity. Controlling the number of hidden nodes is one way of limiting the size of the neural network. On the other hand, since we are generating discrete inputs from a continuous function, the neural network does not have a fixed input size. The number of inputs might affect the performance of the neural network more straightforwardly. As shown in Fig. 2, the contribution from high angular momentum diminishes quickly. This is also reflected in Fig. 3(a) that $l_{\max} = 4$ is sufficient for the angular momentum cutoff. $n_{\max}$ has a similar effect on the overall performance, as seen from Fig. 3(b). Though larger neural networks and denser power spectra can produce smaller cross-validation errors, we choose to continue the investigation of the neural network with $n_{\max} = 10$, $l_{\max} = 4$ and 32 neurons in each hidden layer. In this way, we can achieve a relatively good accuracy with a moderate neural network size (50 input nodes and 2721 parameters in total).

There are two more parameters for input features, $\mu_{\max}$ and c, that need to be optimized. They control the form of radial basis functions and can only be searched by computing the cross-validation errors for a set of values. This process is shown in Fig. S4, and finally $\mu_{\max} = 4$ bohrs and $c = 1.25$ are determined as optimal parameters. Afterward, the neural network will be trained against the entire training set. Because machine learning is intrinsically stochastic, we will generate many sets of neural network parameters and

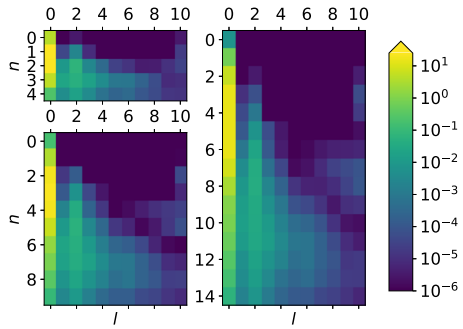

FIG. 2. Superposition of the power spectra computed on the entire database (excluding molecules containing heavy elements), with BLYP densities and parameters $\mu_{\max} = 4$ bohrs, $c = 1.25$. Three figures correspond to $n_{\max} = 5$, 10, and 15, respectively. $l_{\max}$ is set to 10 in all cases, and the power spectra with a smaller $l_{\max}$ is simply a subgraph of the graph with $l_{\max} = 10$.

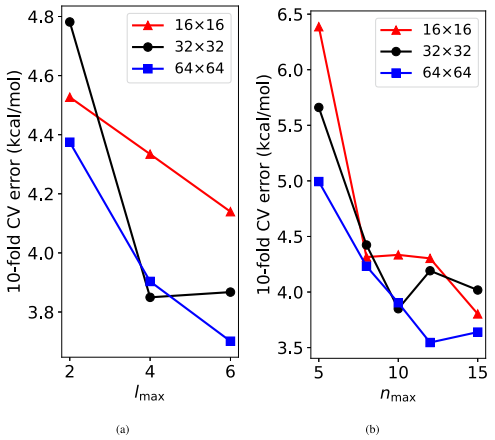

FIG. 3. (a) Cross-validation errors for different l_{max}, with BLYP as the base functional. Other parameters: $n_{max} = 10$, $\mu_{max} = 4$ bohrs, and $c = 1.25$. The size of the neural network is reflected in the labels. For example, 16 × 16 means a neural network with two hidden layers, while each hidden layer has 16 hidden nodes. (b) CV errors for varying n_{max}, while other settings are the same ($l_{max} = 4$).

take the average to reduce the noise. To train this ensemble neural network for the final test, unlike in previous training events, we will first fix the set of D3 parameters. However, they are not fixed at the BLYP-D3(BJ) values. Instead, we first carry out multiple single neural network training tasks to determine this set of D3 parameters. To accomplish this goal, we initiate 20 training tasks (those 10 folds in the CV process are reused as ten different validation sets for early stopping; each split has two tasks associated with different random initial guesses) and determine the set of optimal D3 parameters by observing the smallest combined error from training and validation sets. These new D3 parameters will be used along with the neural network model, and their values can be found in training and evaluation Python scripts for the ensemble neural network (see Data Availability).

Finally, we fix the D3 parameters and perform 50 training tasks (ten splits as before; each split has five different tasks). Consequently, the final evaluation of the machine learning functional comes from the ensemble neural network consisting of an average of 50 neural networks. The standard deviation of predictions generated from each single neural network is usually large. For example, the mean correction for the hydrogen atom is about −0.0019 hartree, which brings the total energy to an accurate value −0.4998 hartree. However, the standard deviation can be as large as 0.0032 hartree, which is even larger than the absolute value of the mean correction. Therefore, the goal of the ensemble evaluation is to use many neural networks to reduce the standard error.

B. Performance evaluation on the test set

To make a fair comparison, the resulting ensemble neural network and corresponding D3 parameters are tested on the reserved pure test set. Since we are using a mixture of MGCDB84 and GMTKN55, here we separate the tests to distinguish where the tests are from and compare the neural network functional with different methods for each database. This is because we want to reuse computed values as provided by these databases, but they did not cover all available functionals, and the best-performing methods ranked by each database are not the same. For tests in the MGCDB84 database, the comparison with ωB97M-V is presented in Table I. Similarly, Table II shows the errors on tests that are from the GMTKN55 database, along with the errors produced by DSD-BLYP-D3(BJ). See Tables S1 and S2 for corresponding training errors. It is intriguing to note that various error indicators [mean absolute deviation (MAD), root of mean squared deviation (RMSD), and WTMAD with scaling factor defined in Eq. (12)] can rank the performance of approximate functionals differently. The RMSD is known to overemphasize large absolute errors, which is the reason why the neural network functional has a large RMSD for the TC category in Table II. For example, the ionization potential (IP) of the oxygen atom produced by BLYP-NN-D3(BJ) has an absolute error of 28.3 kcal/mol, while the relative error is actually only 9%. If this term is absent from the test set, the RMSD of the TC category immediately drops to 4.54 kcal/mol (drops by 20%), while the MAD and the WTMAD are relatively stable. (The MAD becomes 2.87 kcal/mol, which is changed by 12%; the WTMAD becomes 3.24 kcal/mol, which is changed by only 3%.) The MAD is reliable when errors in similar reactions are averaged. It, however, can still ignore reactions with small reaction energies when different datasets are mixed. Therefore, we advocate using the WTMAD to properly report the quality of a

TABLE I. Comparing the neural network functional with ωB97M-V for tests belonging to MGCDB84. TC, thermochemistry reaction energies; BH, barrier heights; IE, isomerization energies; NC, non-covalent interaction energies; RG, rare gas dimmer interaction energies. The unit is kcal/mol.

		MAD	RMSD	WTMAD
TC	BLYP-NN-D3(BJ)	2.59	4.90	2.44
	ωB97M-V	2.62	3.53	3.49
BH	BLYP-NN-D3(BJ)	1.88	2.68	5.14
	ωB97M-V	0.76	0.97	2.06
IE	BLYP-NN-D3(BJ)	0.26	0.61	2.46
	ωB97M-V	0.33	0.49	4.07
NC	BLYP-NN-D3(BJ)	0.17	0.57	2.54
	ωB97M-V	0.15	0.35	3.04
RG	BLYP-NN-D3(BJ)	0.006	0.013	1.57
	ωB97M-V	0.013	0.035	3.29

TABLE II. Comparing the neural network functional with DSD-BLYP-D3(BJ) for tests belong to GMTKN55. Notations are the same as in Table I. The unit is kcal/mol.

		MAD	RMSD	WTMAD
TC	BLYP-NN-D3(BJ)	3.25	5.67	3.33
	DSD-BLYP-D3(BJ)	1.74	2.66	3.28
BH	BLYP-NN-D3(BJ)	1.74	2.73	6.30
	DSD-BLYP-D3(BJ)	1.81	2.73	7.13
IE	BLYP-NN-D3(BJ)	1.45	2.57	4.91
	DSD-BLYP-D3(BJ)	1.06	1.87	4.44
NC	BLYP-NN-D3(BJ)	0.17	0.27	2.44
	DSD-BLYP-D3(BJ)	0.37	0.78	4.56

method when it is tested on a mixture of various different datasets. As seen from the WTMAD values, the neural network functional can at least provide comparable accuracy to the best-performing hybrid and double-hybrid functionals and in many cases actually outperform them. Figure S5 presents the error plot of the neural network functional, along with (double-)hybrid functionals.

We have taken several steps to avoid overfitting in our model. First of all, early stopping is used throughout the entire work; therefore, we can be assured no severe overfitting is going to happen because of too many training cycles. Second, we use K-fold cross-validation scores to determine the proper size of the neural network, and we chose a relatively small one, but still with reasonably good accuracy. Lastly, from the training error (Tables S1 and S2) and final test error (Tables I and II), we can see test errors are not that different from training errors. Therefore, we can draw the conclusion that our model is not overfitted.

Though this neural network functional is intended to be universal, we should be aware that its performance on a particular reaction is highly dependent on the training data fed to the model. We observed that the neural network might give some reaction energies that are worse than those provided by the base method, if the reactions contain elements that do not occur often in the training data. Even for elements that have high occurrence rates in the training data, a few large errors could also be made, such as the large error in the IP of the oxygen atom as pointed out before. The reason for the unsatisfactory performance is because, for electron densities in those systems, they are in regions that are far from where the neural network has been trained with. For the oxygen atom IP issue, one possible explanation is that, molecules containing oxygen included in the training set are mostly bonded molecules. The local density of an ionized O atom differs from what the neural network learned significantly, thus leading to a large error when extrapolating. Even if the neural network does not take the atomic number as an input directly, the local density is mostly determined by the element type. Therefore, before applying this method to systems containing elements that are currently not well explored by the training set used, it is necessary to first extend the training database with more data for those elements.

C. Tests on untrained elements

Next, we study how this functional behaves when faced with elements that are completely untrained. There are reactions in the GMTKN55 database that involve elements heavier than Kr. In addition, the MOR41 database[138] can be used to benchmark the performance on transition metals. The details about the datasets can be summarized as follows:

1. 30 non-covalent interaction energies from the HAL59 set. The untrained element is I.
2. The HEAVY28 set, which includes 28 non-covalent interaction energies that involve Sb, Te, I, Pb, and Bi.
3. 4 reaction energies from the HEAVYSB11 set. The elements of interest are Sn, Sb, Te, and Pb.
4. The MOR41 set, which includes 41 closed-shell organometallic reactions, and the metals in this set are all transition metals (from Ti to Pt).

It is interesting to know how the neural network functional performs on those reactions and the impacts on the machine learning model when part of these data are included in the training set. Table III lists MADs on HAL59, HEAVY28, and HEAVYSB11 for the base functional and the neural network functional that are the model that is trained on light elements only including 50% and 75% of above data in the training. It can be found that, without including any entry in those datasets in the training set, BLYP-NN-D3(BJ) already outperforms BLYP-D3(BJ) in HAL59 and HEAVY28. This is because reaction energies in these two sets are all non-covalent interaction energies. The errors on untrained elements are mostly canceled. Furthermore, including some of them in the training set does improve the accuracy. The HEAVYSB11 set involves chemical reactions, thus the neural network functional can no longer rely on error cancellation. This is also the case for the MOR41 set, which is presented in Table IV. Fortunately, by including part of the data in the training set, the neural network functional in the end can outperform that of the base method. This clearly demonstrates the extensibility of the neural network functional. By supplying more training data on the properties to be measured, the performance of the functional can be systematically improved.

TABLE III. Errors of the neural network functional for 30 non-covalent interaction energies in the HAL59 set, 28 non-covalent interaction energies in the HEAVY28 set, and four reaction energies from the HEAVYSB11 set when some portions of these data are amended to the original training set (indicated by "percentage training"). The extracted four reaction energies from the HEAVYSB11 set constitute an extremely small set such that they are not further split. The percentage under "MAD" shows how large the test set is as compared to the entire dataset. Missing entries indicate that training data should not be used to test the method. For instance, the model trained with 75% data in the dataset can only be tested on the remaining 25%. The unit is kcal/mol.

	HAL59			HEAVY28			HEAVYSB11
	MAD (100%)	MAD (50%)	MAD (25%)	MAD (100%)	MAD (50%)	MAD (25%)	MAD (100%)
BLYP-D3(BJ)	0.81	0.80	0.68	0.35	0.37	0.41	4.26
BLYP-NN-D3(BJ) (0% training)	0.63	0.61	0.52	0.21	0.19	0.13	5.33
BLYP-NN-D3(BJ) (50% training)	⋯	0.25	0.16	⋯	0.14	0.16	⋯
BLYP-NN-D3(BJ) (75% training)	⋯	⋯	0.15	⋯	⋯	0.18	⋯

TABLE IV. Errors of the neural network functional in the MOR41 set when some portions of the dataset are appended to the original training set (indicated by "percentage training"). The percentage accompanying "MAD" shows how large the test set is as compared to the entire dataset. Missing entries indicate that training data should not be used to test the method. For instance, the model trained with 75% data in the dataset can only be tested on the remaining 25%. The unit is kcal/mol.

	MAD (100%)	MAD (50%)	MAD (25%)
BLYP-D3(BJ)	5.24	4.47	2.84
BLYP-NN-D3(BJ) (0% training)	5.53	5.57	4.21
BLYP-NN-D3(BJ) (50% training)	⋯	4.19	3.29
BLYP-NN-D3(BJ) (75% training)	⋯	⋯	2.70

In addition to the effect of the training database, the form of the loss function might also be worth studying. WTMAD2 works best when similar reactions are grouped together. However, existing databases may have entries with quite different magnitudes within a subset. In a subset with a large mean reaction energy ($\overline{|\Delta E|}$), those reactions with small reaction energies will become insignificant when training. In this case, other loss function forms such as the WTMAD1[122] might be more appropriate.

D. Fractional charge behavior of ML functional

Finally, when it comes to studying the fundamental difficulties that traditional DFAs have, which are the fractional charge error and the fractional spin error, because the neural network functional remains an explicit functional in the density, in theory it is still incapable of dealing with these difficulties. In Fig. 4, we show how the energy of the carbon atom behaves for fractional number of electrons with conventional BLYP and with NN-corrected BLYP. It can be found that, as expected, the neural network functional behaves very similarly as compared to the base functional. Because the delocalization error is mostly quadratic in terms of the fractional charge, it is possible that we can apply the local orbital scaling correction (LOSC)[103,104] to reduce delocalization errors. We also plan to incorporate LOSC into the ML process in future functional development, meaning to impose the energy linearity conditions through the LOSC functional form.

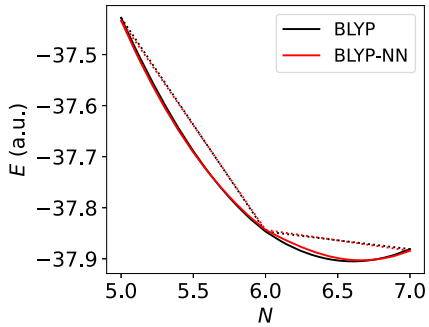

FIG. 4. Energy of a carbon number with varying number of electrons, produced by BLYP and BLYP-NN with def2-TZVP basis.[139] The dotted lines are linear lines connecting the adjacent integer systems, which is the ultimate correct result as required by the Perdew–Parr–Levy–Balduz (PPLB) linearity condition.[95,97]

V. CONCLUSIONS

In conclusion, we have developed an energy correction to existing semi-local functionals. This correction takes the electron density as the input and is expressed as many-body (many-center) expansion, the first term of which is a summation of atomic contributions. This correction is expressed as a universal functional of atom-centered electron densities and is determined by artificial neural networks. Tests have shown that, with a GGA (BLYP) as the base method, this neural network functional can achieve a level of accuracy that is comparable to double-hybrid functionals, while keeping the method computationally affordable. Meanwhile, the tests on heavy elements and transition metals show that this method is possible to be further improved by extending the training dataset with more accurate reference data. This opens up a new direction in the development of nonlocal functionals. Further development could include two-body terms and also the ultra-nonlocal corrections for delocalization and static correlation error. This direction may potentially lead to methods that are of broad interests because of their accuracy for large systems.

SUPPLEMENTARY MATERIAL

Additional tables showing training errors and additional figures about the density partition, radial basis functions, and more machine learning results.

ACKNOWLEDGMENTS

We acknowledge the support from the National Science Foundation (Grant No. CHE-2154831) and National Institutes of Health (Grant No. R01-GM061870).

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Zehua Chen: Data curation (lead); Formal analysis (lead); Investigation (lead); Methodology (equal); Software (lead); Validation (lead); Visualization (lead); Writing – original draft (lead); Writing – review & editing (equal). **Weitao Yang**: Conceptualization (lead); Funding acquisition (lead); Methodology (equal); Project administration (lead); Resources (lead); Supervision (lead); Writing – review & editing (equal).

DATA AVAILABILITY

Modified MGCDB84, GMTKN55 and MOR41 data files; calculated BLYP wave function files; modified `dftd3` program and

DFT-D3(BJ) internal data it generated; Python scripts used to generate power spectra and the power spectra it calculated; training, validation and test data; Python scripts used to train and evaluate the PyTorch model and the resulting models; and the Python script generated the carbon atom energy plot can be found via the Duke Research Data Repository.[140]

REFERENCES

[1] P. Hohenberg and W. Kohn, "Inhomogeneous electron gas," Phys. Rev. **136**, B864–B871 (1964).
[2] W. Kohn and L. J. Sham, "Self-consistent equations including exchange and correlation effects," Phys. Rev. **140**, A1133–A1138 (1965).
[3] U. von Barth and L. Hedin, "A local exchange-correlation potential for the spin polarized case. I," J. Phys. C: Solid State Phys. **5**, 1629 (1972).
[4] S. H. Vosko, L. Wilk, and M. Nusair, "Accurate spin-dependent electron liquid correlation energies for local spin density calculations: A critical analysis," Can. J. Phys. **58**, 1200–1211 (1980).
[5] A. D. Becke, "Density-functional exchange-energy approximation with correct asymptotic behavior," Phys. Rev. A **38**, 3098–3100 (1988).
[6] C. Lee, W. Yang, and R. G. Parr, "Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density," Phys. Rev. B **37**, 785–789 (1988).
[7] J. P. Perdew and Y. Wang, "Accurate and simple analytic representation of the electron-gas correlation energy," Phys. Rev. B **45**, 13244–13249 (1992).
[8] J. P. Perdew, K. Burke, and M. Ernzerhof, "Generalized gradient approximation made simple," Phys. Rev. Lett. **77**, 3865–3868 (1996).
[9] P. Mori-Sánchez, A. J. Cohen, and W. Yang, "Localization and delocalization errors in density functional theory and implications for band-gap prediction," Phys. Rev. Lett. **100**, 146401 (2008).
[10] E. R. Johnson, P. Mori-Sánchez, A. J. Cohen, and W. Yang, "Delocalization errors in density functionals and implications for main-group thermochemistry," J. Chem. Phys. **129**, 204112 (2008).
[11] S. Kristyán and P. Pulay, "Can (semi)local density functional theory account for the London dispersion forces?," Chem. Phys. Lett. **229**, 175–180 (1994).
[12] A. D. Becke, "Density-functional thermochemistry. V. Systematic optimization of exchange-correlation functionals," J. Chem. Phys. **107**, 8554–8560 (1997).
[13] A. D. Becke, "Density-functional thermochemistry. III. The role of exact exchange," J. Chem. Phys. **98**, 5648–5652 (1993).
[14] A. D. Becke, "A new mixing of Hartree–Fock and local density-functional theories," J. Chem. Phys. **98**, 1372–1377 (1993).
[15] S. Grimme, "Semiempirical hybrid density functional with perturbative second-order correlation," J. Chem. Phys. **124**, 034108 (2006).
[16] S. Kozuch, D. Gruzman, and J. M. L. Martin, "DSD-BLYP: A general purpose double hybrid density functional including spin component scaling and dispersion correction," J. Phys. Chem. C **114**, 20801–20808 (2010).
[17] I. Y. Zhang, N. Q. Su, É. A. G. Brémond, C. Adamo, and X. Xu, "Doubly hybrid density functional xDH-PBE0 from a parameter-free global hybrid model PBE0," J. Chem. Phys. **136**, 174103 (2012).
[18] N. Q. Su and X. Xu, "Construction of a parameter-free doubly hybrid density functional from adiabatic connection," J. Chem. Phys. **140**, 18A512 (2014).
[19] N. Mardirossian and M. Head-Gordon, "Thirty years of density functional theory in computational chemistry: An overview and extensive assessment of 200 density functionals," Mol. Phys. **115**, 2315–2372 (2017).
[20] Q. Wu and W. Yang, "Empirical correction to density functional theory for van der Waals interactions," J. Chem. Phys. **116**, 515–524 (2002).
[21] E. Hult, Y. Andersson, B. I. Lundqvist, and D. C. Langreth, "Density functional for van der Waals forces at surfaces," Phys. Rev. Lett. **77**, 2029–2032 (1996).
[22] M. Dion, H. Rydberg, E. Schröder, D. C. Langreth, and B. I. Lundqvist, "Van der Waals density functional for general geometries," Phys. Rev. Lett. **92**, 246401 (2004).
[23] A. D. Becke and E. R. Johnson, "A density-functional model of the dispersion interaction," J. Chem. Phys. **123**, 154101 (2005).
[24] A. D. Becke and E. R. Johnson, "Exchange-hole dipole moment and the dispersion interaction: High-order dispersion coefficients," J. Chem. Phys. **124**, 014104 (2006).
[25] A. Tkatchenko and M. Scheffler, "Accurate molecular van der Waals interactions from ground-state electron density and free-atom reference data," Phys. Rev. Lett. **102**, 073005 (2009).
[26] O. A. Vydrov and T. Van Voorhis, "Nonlocal van der Waals density functional: The simpler the better," J. Chem. Phys. **133**, 244103 (2010).
[27] R. A. DiStasio, O. A. von Lilienfeld, and A. Tkatchenko, "Collective many-body van der Waals interactions in molecular systems," Proc. Natl. Acad. Sci. U. S. A. **109**, 14791–14795 (2012).
[28] K. Berland, V. R. Cooper, K. Lee, E. Schröder, T. Thonhauser, P. Hyldgaard, and B. I. Lundqvist, "van der Waals forces in density functional theory: A review of the vdW-DF method," Rep. Prog. Phys. **78**, 066501 (2015).
[29] A. Tkatchenko, R. A. DiStasio, Jr., R. Car, and M. Scheffler, "Accurate and efficient method for many-body van der Waals interactions," Phys. Rev. Lett. **108**, 236402 (2012).
[30] A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Žídek, A. W. R. Nelson, A. Bridgland, H. Penedones, S. Petersen, K. Simonyan, S. Crossan, P. Kohli, D. T. Jones, D. Silver, K. Kavukcuoglu, and D. Hassabis, "Improved protein structure prediction using potentials from deep learning," Nature **577**, 706–710 (2020).
[31] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis, "Highly accurate protein structure prediction with AlphaFold," Nature **596**, 583–589 (2021).
[32] J. Behler and M. Parrinello, "Generalized neural-network representation of high-dimensional potential-energy surfaces," Phys. Rev. Lett. **98**, 146401 (2007).
[33] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, "Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons," Phys. Rev. Lett. **104**, 136403 (2010).
[34] L. Zhang, J. Han, H. Wang, R. Car, and W. E, "Deep potential molecular dynamics: A scalable model with the accuracy of quantum mechanics," Phys. Rev. Lett. **120**, 143001 (2018).
[35] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, "Fast and accurate modeling of molecular atomization energies with machine learning," Phys. Rev. Lett. **108**, 058301 (2012).
[36] K. Hansen, G. Montavon, F. Biegler, S. Fazli, M. Rupp, M. Scheffler, O. A. Von Lilienfeld, A. Tkatchenko, and K.-R. Müller, "Assessment and validation of machine learning methods for predicting molecular atomization energies," J. Chem. Theory Comput. **9**, 3404–3419 (2013).
[37] 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).
[38] X. Wang, Y. Xu, H. Zheng, and K. Yu, "A scalable graph neural network method for developing an accurate force field of large flexible organic molecules," J. Phys. Chem. Lett. **12**, 7982–7987 (2021).
[39] S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky, "E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials," Nat. Commun. **13**, 2453 (2022).
[40] Z. Cheng, J. Du, L. Zhang, J. Ma, W. Li, and S. Li, "Building quantum mechanics quality force fields of proteins with the generalized energy-based fragmentation approach and machine learning," Phys. Chem. Chem. Phys. **24**, 1326–1337 (2022).
[41] A. P. Latham and B. Zhang, "Unifying coarse-grained force fields for folded and disordered proteins," Curr. Opin. Struct. Biol. **72**, 63–70 (2022).
[42] O. T. Unke, S. Chmiela, H. E. Sauceda, M. Gastegger, I. Poltavsky, K. T. Schütt, A. Tkatchenko, and K.-R. Müller, "Machine learning force fields," Chem. Rev. **121**, 10142–10186 (2021).
[43] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. D. Sawaya, S. Sim, L. Veis, and A. Aspuru-Guzik, "Quantum chemistry in the age of quantum computing," Chem. Rev. **119**, 10856–10915 (2019).

[44]T. W. Ko, J. A. Finkler, S. Goedecker, and J. Behler, "General-purpose machine learning potentials capturing nonlocal charge transfer," Acc. Chem. Res. **54**, 808–817 (2021).
[45]I. Poltavsky and A. Tkatchenko, "Machine learning force fields: Recent advances and remaining challenges," J. Phys. Chem. Lett. **12**, 6551–6564 (2021).
[46]P. O. Dral, *Quantum Chemistry in the Age of Machine Learning* (Elsevier, 2022).
[47]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**, 2087–2096 (2015).
[48]D. J. Tozer, V. E. Ingamells, and N. C. Handy, "Exchange-correlation potentials," J. Chem. Phys. **105**, 9200–9213 (1996).
[49]X. Zheng, L. Hu, X. Wang, and G. Chen, "A generalized exchange-correlation functional: The neural-networks approach," Chem. Phys. Lett. **390**, 186–192 (2004).
[50]J. C. Snyder, M. Rupp, K. Hansen, K.-R. Müller, and K. Burke, "Finding density functionals with machine learning," Phys. Rev. Lett. **108**, 253002 (2012).
[51]M. Alghadeer, A. Al-Aswad, and F. H. Alharbi, "Highly accurate machine learning model for kinetic energy density functional," Phys. Lett. A **414**, 127621 (2021).
[52]F. Imoto, M. Imada, and A. Oshiyama, "Order-*N* orbital-free density-functional calculations with machine learning of functional derivatives for semiconductors and metals," Phys. Rev. Res. **3**, 033198 (2021).
[53]J. Lee, A. Seko, K. Shitara, K. Nakayama, and I. Tanaka, "Prediction model of band gap for inorganic compounds by combination of density functional theory calculations and machine learning techniques," Phys. Rev. B **93**, 115104 (2016).
[54]G. Pilania, J. E. Gubernatis, and T. Lookman, "Multi-fidelity machine learning models for accurate bandgap predictions of solids," Comput. Mater. Sci. **129**, 156–163 (2017).
[55]J. R. Moreno, J. Flick, and A. Georges, "Machine learning band gaps from the electron density," Phys. Rev. Mater. **5**, 083802 (2021).
[56]J. Gedeon, J. Schmidt, M. J. P. Hodgson, J. Wetherell, C. L. Benavides-Riveros, and M. A. L. Marques, "Machine learning the derivative discontinuity of density-functional theory," Mach. Learn.: Sci. Technol. **3**, 015011 (2021).
[57]F. Brockherde, L. Vogt, L. Li, M. E. Tuckerman, K. Burke, and K.-R. Müller, "Bypassing the Kohn–Sham equations with machine learning," Nat. Commun. **8**, 872 (2017).
[58]A. Chandrasekaran, D. Kamal, R. Batra, C. Kim, L. Chen, and R. Ramprasad, "Solving the electronic structure problem with machine learning," npj Comput. Mater. **5**, 22 (2019).
[59]J. A. Ellis, L. Fiedler, G. A. Popoola, N. A. Modine, J. A. Stephens, A. P. Thompson, A. Cangi, and S. Rajamanickam, "Accelerating finite-temperature Kohn–Sham density functional theory with deep neural networks," Phys. Rev. B **104**, 035120 (2021).
[60]B. Focassio, M. Domina, U. Patil, A. Fazzio, and S. Sanvito, "Linear Jacobi–Legendre expansion of the charge density for machine learning-accelerated electronic structure calculations," npj Comput. Mater. **9**, 87 (2023).
[61]L. Fiedler, N. A. Modine, S. Schmerler, D. J. Vogel, G. A. Popoola, A. P. Thompson, S. Rajamanickam, and A. Cangi, "Predicting electronic structures at any length scale with machine learning," npj Comput. Mater. **9**, 115 (2023).
[62]X. Shao, L. Paetow, M. E. Tuckerman, and M. Pavanello, "Machine learning electronic structure methods based on the one-electron reduced density matrix," Nat. Commun. **14**, 6281 (2023).
[63]G. Hegde and R. C. Bowen, "Machine-learned approximations to density functional theory Hamiltonians," Sci. Rep. **7**, 42669 (2017).
[64]H. Li, Z. Wang, N. Zou, M. Ye, R. Xu, X. Gong, W. Duan, and Y. Xu, "Deep-learning density functional theory Hamiltonian for efficient *ab initio* electronic-structure calculation," Nat. Comput. Sci. **2**, 367–377 (2022).
[65]H. Ji and Y. Jung, "A local environment descriptor for machine-learned density functional theory at the generalized gradient approximation level," J. Chem. Phys. **148**, 241742 (2018).
[66]R. Nagai, R. Akashi, S. Sasaki, and S. Tsuneyuki, "Neural-network Kohn–Sham exchange-correlation potential and its out-of-training transferability," J. Chem. Phys. **148**, 241737 (2018).
[67]Y. Zhou, J. Wu, S. Chen, and G. Chen, "Toward the exact exchange–correlation potential: A three-dimensional convolutional neural network construct," J. Phys. Chem. Lett. **10**, 7264–7269 (2019).
[68]Y. Suzuki, R. Nagai, and J. Haruyama, "Machine learning exchange-correlation potential in time-dependent density-functional theory," Phys. Rev. A **101**, 050501 (2020).
[69]A. Grisafi, A. Fabrizio, B. Meyer, D. M. Wilkins, C. Corminboeuf, and M. Ceriotti, "Transferable machine-learning model of the electron density," ACS Cent. Sci. **5**, 57–64 (2019).
[70]M. Bogojeski, L. Vogt-Maranto, M. E. Tuckerman, K.-R. Müller, and K. Burke, "Quantum chemical accuracy from density functional approximations via machine learning," Nat. Commun. **11**, 5223 (2020).
[71]Q. Xu, C. Ma, W. Mi, Y. Wang, and Y. Ma, "Nonlocal pseudopotential energy density functional for orbital-free density functional theory," Nat. Commun. **13**, 1385 (2022).
[72]Y. Bai, L. Vogt-Maranto, M. E. Tuckerman, and W. J. Glover, "Machine learning the Hohenberg–Kohn map for molecular excited states," Nat. Commun. **13**, 7044 (2022).
[73]D. Golze, M. Hirvensalo, P. Hernández-León, A. Aarva, J. Etula, T. Susi, P. Rinke, T. Laurila, and M. A. Caro, "Accurate computational prediction of core-electron binding energies in carbon-based materials: A machine-learning model combining density-functional theory and GW," Chem. Mater. **34**, 6240–6254 (2022).
[74]R. Nagai, R. Akashi, and O. Sugino, "Completing density functional theory by machine learning hidden messages from molecules," npj Comput. Mater. **6**, 43 (2020).
[75]J. T. Margraf and K. Reuter, "Pure non-local machine-learned density functional theory for electron correlation," Nat. Commun. **12**, 344 (2021).
[76]J. Kirkpatrick, B. McMorrow, D. H. P. Turban, A. L. Gaunt, J. S. Spencer, A. G. D. G. Matthews, A. Obika, L. Thiry, M. Fortunato, D. Pfau, L. R. Castellanos, S. Petersen, A. W. R. Nelson, P. Kohli, P. Mori-Sánchez, D. Hassabis, and A. J. Cohen, "Pushing the frontiers of density functionals by solving the fractional electron problem," Science **374**, 1385–1389 (2021).
[77]K. Bystrom and B. Kozinsky, "CIDER: An expressive, nonlocal feature set for machine learning density functionals with exact constraints," J. Chem. Theory Comput. **18**, 2180–2192 (2022).
[78]S. Dick and M. Fernandez-Serra, "Machine learning accurate exchange and correlation functionals of the electronic density," Nat. Commun. **11**, 3509 (2020).
[79]X. Lei and A. J. Medford, "Design and analysis of machine learning exchange-correlation functionals via rotationally invariant convolutional descriptors," Phys. Rev. Mater. **3**, 063801 (2019).
[80]Y. Chen, L. Zhang, H. Wang, and W. E, "DeePKS: A comprehensive data-driven approach toward chemically accurate density functional theory," J. Chem. Theory Comput. **17**, 170–181 (2020).
[81]Y. Chen, L. Zhang, H. Wang, and W. E, "DeePKS-kit: A package for developing machine learning-based chemically accurate energy and density functional models," Comput. Phys. Commun. **282**, 108520 (2023).
[82]O. A. von Lilienfeld and K. Burke, "Retrospective on a decade of machine learning for chemical discovery," Nat. Commun. **11**, 4895 (2020).
[83]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).
[84]R. Pederson, B. Kalita, and K. Burke, "Machine learning and density functional theory," Nat. Rev. Phys. **4**, 357–358 (2022).
[85]J. Wu, G. Chen, J. Wang, and X. Zheng, "Chapter 23—Redesigning density functional theory with machine learning," in *Quantum Chemistry in the Age of Machine Learning*, edited by P. O. Dral (Elsevier, 2023), pp. 531–558.
[86]J. Behler, "Atom-centered symmetry functions for constructing high-dimensional neural network potentials," J. Chem. Phys. **134**, 074106 (2011).
[87]T. W. Ko, J. A. Finkler, S. Goedecker, and J. Behler, "A fourth-generation high-dimensional neural network potential with accurate electrostatics including non-local charge transfer," Nat. Commun. **12**, 398 (2021).
[88]L. Shen, J. Wu, and W. Yang, "Multiscale quantum mechanics/molecular mechanics simulations with neural networks," J. Chem. Theory Comput. **12**, 4934–4946 (2016).
[89]L. Shen and W. Yang, "Molecular dynamics simulations with quantum mechanics/molecular mechanics and adaptive neural networks," J. Chem. Theory Comput. **14**, 1442–1455 (2018).

[90]H. Wang and W. Yang, "Force field for water based on neural network," J. Phys. Chem. Lett. **9**, 3232–3240 (2018).
[91]H. Wang and W. Yang, "Toward building protein force fields by residue-based systematic molecular fragmentation and neural network," J. Chem. Theory Comput. **15**, 1409–1417 (2019).
[92]P. Zhang and W. Yang, "Toward a general neural network force field for protein simulations: Refining the intramolecular interaction in protein," J. Chem. Phys. **159**, 024118 (2023).
[93]D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow, "Harmonic networks: Deep translation and rotation equivariance," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5028–5037, https://openaccess.thecvf.com/content_cvpr_2017/html/Worrall_Harmonic_Networks_Deep_CVPR_2017_paper.html. (2017).
[94]J. Applequist, "Maxwell–Cartesian spherical harmonics in multipole potentials and atomic orbitals," Theor. Chim. Acta **107**, 103–115 (2002).
[95]J. Perdew, R. Parr, M. Levy, and J. Balduz, "Density-functional theory for fractional particle number: Derivative discontinuities of the energy," Phys. Rev. Lett. **49**, 1691–1694 (1982).
[96]A. J. Cohen, P. Mori-Sánchez, and W. Yang, "Fractional spins and static correlation error in density functional theory," J. Chem. Phys. **129**, 121104 (2008).
[97]W. Yang, Y. Zhang, and P. W. Ayers, "Degenerate ground states and a fractional number of electrons in density and reduced density matrix functional theory," Phys. Rev. Lett. **84**, 5172 (2000).
[98]P. Mori-Sánchez, A. J. Cohen, and W. Yang, "Discontinuous nature of the exchange-correlation functional in strongly correlated systems," Phys. Rev. Lett. **102**, 066403 (2009).
[99]A. J. Cohen, P. Mori-Sánchez, and W. Yang, "Fractional view of the exchange-correlation functional and derivative discontinuity in density functional theory," Psi-k Scientific Highlight of the Month 99 (2010).
[100]A. J. Cohen, P. Mori-Sánchez, and W. Yang, "Insights into current limitations of density functional theory," Science **321**, 792–794 (2008).
[101]Y. Andersson, D. C. Langreth, and B. I. Lundqvist, "van der Waals interactions in density-functional theory," Phys. Rev. Lett. **76**, 102–105 (1996).
[102]A. D. Becke and E. R. Johnson, "Exchange-hole dipole moment and the dispersion interaction," J. Chem. Phys. **122**, 154104 (2005).
[103]C. Li, X. Zheng, N. Q. Su, and W. Yang, "Localized orbital scaling correction for systematic elimination of delocalization error in density functional approximations," Natl. Sci. Rev. **5**, 203–215 (2018).
[104]N. Q. Su, C. Li, and W. Yang, "Describing strong correlation with fractional-spin correction in density functional theory," Proc. Natl. Acad. Sci. U. S. A. **115**, 9678–9683 (2018).
[105]M. S. Gordon, D. G. Fedorov, S. R. Pruitt, and L. V. Slipchenko, "Fragmentation methods: A Route to accurate calculations on large systems," Chem. Rev. **112**, 632–672 (2012).
[106]R. M. Richard, K. U. Lao, and J. M. Herbert, "Aiming for benchmark accuracy with the many-body expansion," Acc. Chem. Res. **47**, 2828–2836 (2014).
[107]M. A. Collins and R. P. A. Bettens, "Energy-based molecular fragmentation methods," Chem. Rev. **115**, 5607–5642 (2015).
[108]H. Stoll, "Correlation energy of diamond," Phys. Rev. B **46**, 6700–6704 (1992).
[109]B. Paulus, "The method of increments—A wavefunction-based *ab initio* correlation method for solids," Phys. Rep. **428**, 1–52 (2006).
[110]F. L. Hirshfeld, "XVII. Spatial partitioning of charge density," Isr. J. Chem. **16**, 198–201 (1977).
[111]B. Miehlich, A. Savin, H. Stoll, and H. Preuss, "Results obtained with the correlation energy density functionals of Becke and Lee, Yang and Parr," Chem. Phys. Lett. **157**, 200–206 (1989).
[112]S. Grimme, J. Antony, S. Ehrlich, and H. Krieg, "A consistent and accurate *ab initio* parametrization of density functional dispersion correction (DFT-D) for the 94 elements H-Pu," J. Chem. Phys. **132**, 154104 (2010).
[113]S. Grimme, S. Ehrlich, and L. Goerigk, "Effect of the damping function in dispersion corrected density functional theory," J. Comput. Chem. **32**, 1456–1465 (2011).
[114]M. Welborn, L. Cheng, and T. F. Miller, "Transferability in machine learning for electronic structure via the molecular orbital basis," J. Chem. Theory Comput. **14**, 4772–4779 (2018).
[115]J. P. Coe, "Machine learning configuration interaction," J. Chem. Theory Comput. **14**, 5739–5749 (2018).
[116]L. Hu, X. Wang, L. Wong, and G. Chen, "Combined first-principles calculation and neural-network correction approach for heat of formation," J. Chem. Phys. **119**, 11501–11507 (2003).
[117]Y. Zhao and D. G. Truhlar, "A new local density functional for main-group thermochemistry, transition metal bonding, thermochemical kinetics, and noncovalent interactions," J. Chem. Phys. **125**, 194101 (2006).
[118]Y. Zhao and D. G. Truhlar, "The M06 suite of density functionals for main group thermochemistry, thermochemical kinetics, noncovalent interactions, excited states, and transition elements: Two new functionals and systematic testing of four M06-class functionals and 12 other functionals," Theor. Chem. Acc. **120**, 215–241 (2008).
[119]N. Mardirossian and M. Head-Gordon, "ωB97X-V: A 10-parameter, range-separated hybrid, generalized gradient approximation density functional with nonlocal correlation, designed by a survival-of-the-fittest strategy," Phys. Chem. Chem. Phys. **16**, 9904–9924 (2014).
[120]N. Mardirossian and M. Head-Gordon, "ωB97M-V: A combinatorially optimized, range-separated hybrid, meta-GGA density functional with VV10 nonlocal correlation," J. Chem. Phys. **144**, 214110 (2016).
[121]A. P. Bartók, R. Kondor, and G. Csányi, "On representing chemical environments," Phys. Rev. B **87**, 184115 (2013).
[122]L. Goerigk, A. Hansen, C. Bauer, S. Ehrlich, A. Najibi, and S. Grimme, "A look at the density functional theory zoo with the advanced GMTKN55 database for general main group thermochemistry, kinetics and noncovalent interactions," Phys. Chem. Chem. Phys. **19**, 32184–32215 (2017).
[123]K. Raghavachari, G. W. Trucks, J. A. Pople, and M. Head-Gordon, "A fifth-order perturbation comparison of electron correlation theories," Chem. Phys. Lett. **157**, 479–483 (1989).
[124]R. J. Bartlett and M. Musiał, "Coupled-cluster theory in quantum chemistry," Rev. Mod. Phys. **79**, 291 (2007).
[125]A. Karton, E. Rabinovich, J. M. Martin, and B. Ruscic, "W4 theory for computational thermochemistry: In pursuit of confident sub-kJ/mol predictions," J. Chem. Phys. **125**, 144108 (2006).
[126]A. Karton, P. R. Taylor, and J. M. Martin, "Basis set convergence of post-ccsd contributions to molecular atomization energies," J. Chem. Phys. **127**, 064104 (2007).
[127]A. Karton, S. Daon, and J. M. Martin, "W4-11: A high-confidence benchmark dataset for computational thermochemistry derived from first-principles W4 data," Chem. Phys. Lett. **510**, 165–178 (2011).
[128]S. J. Chakravorty, S. R. Gwaltney, E. R. Davidson, F. A. Parpia, and C. F. p Fischer, "Ground-state correlation energies for atomic ions with 3 to 18 electrons," Phys. Rev. A **47**, 3649 (1993).
[129]M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, G. A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A. V. Marenich, J. Bloino, B. G. Janesko, R. Gomperts, B. Mennucci, H. P. Hratchian, J. V. Ortiz, A. F. Izmaylov, J. L. Sonnenberg, D. Williams-Young, F. Ding, F. Lipparini, F. Egidi, J. Goings, B. Peng, A. Petrone, T. Henderson, D. Ranasinghe, V. G. Zakrzewski, J. Gao, N. Rega, G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, K. Throssell, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. J. Bearpark, J. J. Heyd, E. N. Brothers, K. N. Kudin, V. N. Staroverov, T. A. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A. P. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, J. M. Millam, M. Klene, C. Adamo, R. Cammi, J. W. Ochterski, R. L. Martin, K. Morokuma, O. Farkas, J. B. Foresman, and D. J. Fox, *Gaussian 16 Revision C.01*, Gaussian, Inc., Wallingford CT, 2016.
[130]D. Rappoport and F. Furche, "Property-optimized Gaussian basis sets for molecular response calculations," J. Chem. Phys. **133**, 134105 (2010).
[131]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.-L. Chan, "PySCF: The Python-based simulations of chemistry framework," Wiley Interdiscip. Rev.: Comput. Mol. Sci. **8**, e1340 (2018).
[132]Q. Sun, X. Zhang, S. Banerjee, P. Bao, M. Barbry, N. S. Blunt, N. A. Bogdanov, G. H. Booth, J. Chen, Z.-H. Cui, J. J. Eriksen, Y. Gao, S. Guo, J. Hermann, M. R. Hermes, K. Koh, P. Koval, S. Lehtola, Z. Li, J. Liu, N. Mardirossian, J. D. McClain, M. Motta, B. Mussard, H. Q. Pham, A. Pulkin, W. Purwanto, P. J. Robinson, E.

Ronca, E. R. Sayfutyarova, M. Scheurer, H. F. Schurkus, J. E. T. Smith, C. Sun, S.-N. Sun, S. Upadhyay, L. K. Wagner, X. Wang, A. White, J. D. Whitfield, M. J. Williamson, S. Wouters, J. Yang, J. M. Yu, T. Zhu, T. C. Berkelbach, S. Sharma, A. Y. Sokolov, and G. K.-L. Chan, "Recent developments in the PySCF program package," J. Chem. Phys. **153**, 024109 (2020).

[133]See https://github.com/dftbplus/dftd3-lib for more information about Library version of S. Grimmes DFTD3 code.

[134]A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems* (Curran Associates, 2019), Vol. 32, https://proceedings.neurips.cc/paper_files/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

[135]D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980 (2014).

[136]K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision* (IEEE, 2015), pp. 1026–1034.

[137]L. Li, S. Hoyer, R. Pederson, R. Sun, E. D. Cubuk, P. Riley, K. Burke *et al.*, "Kohn-Sham equations as regularizer: Building prior knowledge into machine-learned physics," Phys. Rev. Lett. **126**, 036401 (2021).

[138]S. Dohm, A. Hansen, M. Steinmetz, S. Grimme, and M. P. Checinski, "Comprehensive thermochemical benchmark set of realistic closed-shell metal organic reactions," J. Chem. Theory Comput. **14**, 2596–2608 (2018).

[139]F. Weigend and R. Ahlrichs, "Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy," Phys. Chem. Chem. Phys. **7**, 3297–3305 (2005).

[140]See https://doi.org/10.7924/r4fj2p230 for more information about Duke Research Data Repository.