CHAPTER

23

Redesigning density functional theory with machine learning

Jiang Wu[a], *Guanhua Chen*[a,*], *Jingchun Wang*[b], *and Xiao Zheng*[b]

[a]Hong Kong Quantum AI Lab and Department of Chemistry, The University of Hong Kong, Pokfulam, Hong Kong [b]Department of Chemical Physics, University of Science and Technology of China, Hefei, China

*Corresponding author: E-mail address: ghc.hku@gmail.com

Abstract

The exchange–correlation (XC) functional plays the central role in density functional theory (DFT). The exact XC functional determines a unique and universal mapping from the electron density of a system to either the XC potential or the XC energy/energy density. Through a self-consistent way, all properties of the system can be calculated by the mapping. The exact XC functional is hard to find, and various popular approximations to it struggled to improve the accuracy further by traditional means throughout the past few decades. In this chapter, we will review several approaches that redesign the DFT XC functional by machine learning (ML) (ML-DFTXC). We will start from one of the earliest functional models that use the global density directly and then move forward to more recent models that were built around the quasi-local electron density, elaborating them with concrete examples. Being the focus of this chapter, the section for quasi-local ML-DFTXC models will be introduced from a solid theoretical foundation, the holographic electron density theorem, and be concluded with a general framework that encompasses all existing models. Auxiliary ML models for van der Waals interactions, which can be added on top of the quasi-local models, will also be discussed. For the tutorial section, an open-source code and related examples will be provided.

Keywords: Exchange-correlation functional/potential, Machine learning, Deep learning, Convolution neural network, Quasi-local electron density, Density functional theory

Introduction

Redesigning the exchange–correlation (XC) functional of density functional theory (DFT) with machine learning (ML) algorithms, or ML-DFTXC, started as early as Handy and

co-workers' work in 1996 [1]. Later, the concept was rediscovered independently by Zheng and co-workers in 2004 [2]. Since then, several proposals have been made to construct the XC functional with various ML methods. In particular, over the past few years, several methods have been developed aiming at reproducing the accurate electron density, instead of reproducing energy-related scalar benchmarks only. In this chapter, the development of the ML method for the DFT XC functional will be reviewed and discussed from the inception of ML-DFTXC [1,2] to the era of deep learning [3–6].

Characterized by a *deterministic* and *unique* mapping between electron density and XC energy or potential, the philosophy of integrating ML into DFT is fundamentally different from statistical regressions, where information is usually immersed in random noises. On the other hand, the deterministic mapping in DFT is so complicated, which goes far beyond any ideas of mere parameter fitting. Fortunately, the task of finding such a deterministic mapping shares certain fundamental similarity with a series of already successful stories completed by ML, from which ML-DFTXC can borrow conceptually. Surprisingly in hindsight, modern deep learning algorithms quickly become able to recognize faces, fold a protein [7], and beat humans in their own games [8] alike. Motivated by the vast applications in the field of computer vision and natural language processing, more efficient deep learning algorithms are becoming available, including convolutional neural network (CNN) [9], graph neural network (GNN) [10], and transformers [11]. In the era of deep learning, a new path toward the heaven of chemical accuracy [12] is unfolding in front of us.

The existing efforts on the subject of ML-DFTXC come in different aspects and can be roughly classified into several categories: the ML-redesigned XC potential mapping [3,13], the ML-redesigned XC energy functional mappings [2,4,5,14–23], ML-redesigned fragment XC energy functional mappings [6], and other categories beyond [24–35]. It is worth mentioning that except for the XC functionals/potentials, other parts of the DFT framework have also been targeted and improved by ML, among which the kinetic energy functional attracted broad attention [36–43]. ML has also been extensively applied to fitting or constructing potential energy surfaces (e.g., [44]), where DFT is often used as either a training target or one of the benchmarks for the ML algorithms. In this type of works, DFT's role can be replaced with various other quantum chemistry methods as well. We would like to clarify that the current chapter does not attempt to be as thorough as possible. On the other hand, it is intended to only cover ML-DFTXC and focuses on methods that actually 'redesign' DFT's XC functional or potential mapping with ML, as suggested by the title.

Methods

Global electron density formulation of ML-DFTXC

Dictated by the Hohenberg–Kohn theorem [45], the ground-state electron density contains all the information of a given system. Therefore, one can always construct an ML-DFTXC formulation starting from the global electron density function of the entire system. The XC energy E_{XC}, as well as the XC potential at any position in real space $v_{XC}(\boldsymbol{r})$, can be deduced from the global electron density. To describe and formulate such a complicated correspondence from any physical electron density to its corresponding XC energy or potential, a hybrid functional from traditional DFT provides a sensible starting point.

As one of the most popular hybrid functionals, the B3LYP [46] functional includes five pure functional terms, $E_X^{Slater}[\rho]$, $E_X^{HF}[\rho]$, $\Delta E_X^{Becke}[\rho]$, $E_C^{LYP}[\rho]$, and $E_C^{VWN}[\rho]$, which are tuned by three coefficients, a_0, a_X, and a_C. $\Delta E_X^{Becke}[\rho]$ is the difference between the Becke88 exchange functional [47] and the Slater [48] exchange functional $E_X^{B88}[\rho] - E_X^{Slater}[\rho]$. The other terms, $E_X^{Slater}[\rho]$, $E_X^{HF}[\rho]$, $E_C^{LYP}[\rho]$, and $E_C^{VWN}[\rho]$, are the Slater exchange, Hartree–Fock (HF) exchange [49], Lee–Yang–Parr (LYP) [46] correlation, and Vosko–Wilk-Nusair (VWN) [50] correlation functionals, respectively.

$$E_{XC}^{B3LYP} = a_0 E_X^{Slater}[\rho] + (1-a_0)E_X^{HF}[\rho] + a_X \Delta E_X^{Becke}[\rho] + a_C E_C^{LYP}[\rho] + (1-a_C)E_C^{VWN}[\rho]. \quad (1)$$

The coefficients in a hybrid functional are fitted to experiments / more accurate calculations and fixed as constants afterward. In B3LYP, the values of the three coefficients ($a_0 = 0.8$, $a_X = 0.72$, and $a_C = 0.81$) came directly from Becke's original fitting to a set of atomization energies, ionization potentials, and so on. [47].

Keen readers may realize that in theory, the coefficients should be system-dependent too. Indeed, if one also writes these coefficients as density functionals (any system dependency is density dependency by the Hohenberg–Kohn [45] theorem), the expression becomes an exact density functional:

$$\begin{aligned} E_{XC}^{Exact} = {} & a_0[\rho]E_X^{Slater}[\rho] + (1-a_0[\rho])E_X^{HF}[\rho] + a_X[\rho]\Delta E_X^{Becke}[\rho] \\ & + a_C[\rho]E_C^{LYP}[\rho] + (1-a_C[\rho])E_C^{VWN}[\rho]. \end{aligned} \quad (2)$$

Being cast in this form, the coefficients become system-dependent too, that is, a_0, a_X, and a_C are different for different systems as the electron density functions ρ are different. All the physical information that the individual pure functionals failed to capture are cast in the functional forms of the coefficients. Therefore, learning the exact density functional becomes equivalent to learning the density functionals of these coefficients $a_0[\rho]$, $a_X[\rho]$, and $a_C[\rho]$.

A milestone was laid in 2004, when the first attempt [2] was made to learn this exact XC functional. The work cast the exact XC functional in the above form and trained a neural network for $a_0[\rho]$, $a_X[\rho]$, and $a_C[\rho]$. The neural network has five descriptors as the inputs and two neurons as the hidden layer. The five descriptors are the functionals of electron density. The detail of the neural network structure is shown in Fig. 1. The neural network-predicted coefficients are then combined with the pure functionals by (2), and the whole XC functional is used in Kohn–Sham (KS) self-consistent field (SCF) [51] calculations.

The XC potential v_{XC} is needed for the KS SCF calculation [51]. When calculating the XC potential from the XC energy (2), a functional derivative of the energy expression is taken with respect to $\rho(\boldsymbol{r})$ (at every position $\boldsymbol{r}$):

$$\begin{aligned} v_{XC}^{Exact}(\mathtt{r}) &= \frac{\delta E_{XC}^{Exact}}{\delta \rho(\mathtt{r})} \\ &= \underbrace{a_0 \frac{\delta E_C^{Slater}}{\delta \rho(\mathtt{r})} + (1-a_0)\frac{\delta E_X^{HF}}{\delta \rho(\mathtt{r})} + a_X \frac{\delta \Delta E_X^{Becke}}{\delta \rho(\mathtt{r})} + a_C \frac{\delta E_C^{LYP}}{\delta \rho(\mathtt{r})} + (1-a_C)\frac{\delta E_C^{VWN}}{\delta \rho(\mathtt{r})}}_{\text{terms containing partial derivatives w.r.t energy functionals}} \\ &\quad + \underbrace{\frac{\delta a_0}{\delta \rho(\mathtt{r})}E_C^{Slater} - \frac{\delta a_0}{\delta \rho(\mathtt{r})}E_X^{HF} + \frac{\delta a_X}{\delta \rho(\mathtt{r})}\Delta E_X^{Becke} + \frac{\delta a_C}{\delta \rho(\mathtt{r})}E_C^{LYP} - \frac{\delta a_C}{\delta \rho(\mathtt{r})}E_C^{VWN}}_{\text{terms contraining partial derivatives w.r.t } a_0, a_X, \text{and } a_C}. \end{aligned} \quad (3)$$

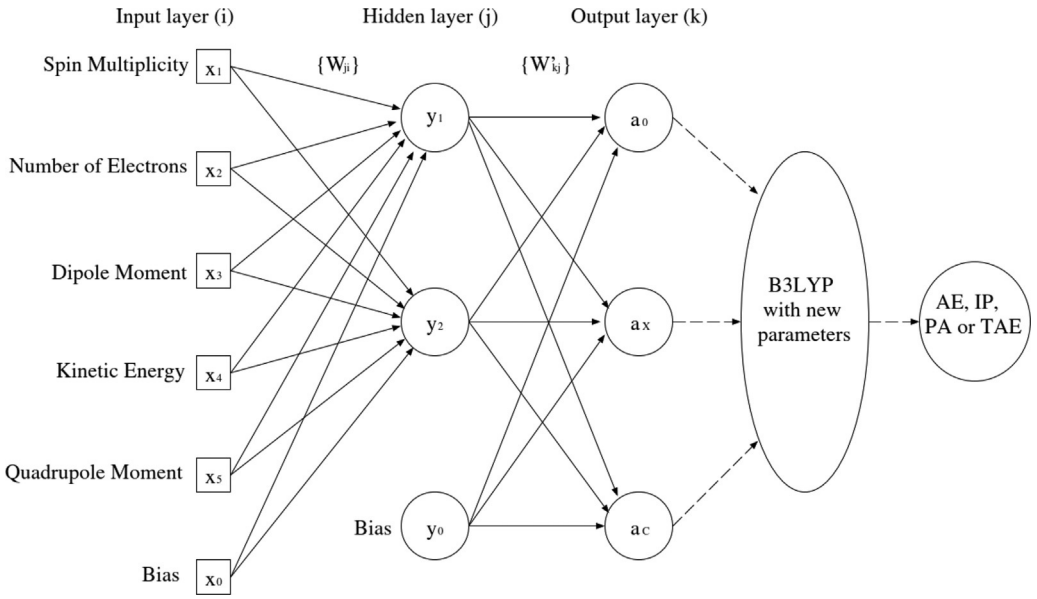

FIG. 1 The structure of a neural network that learns the system-dependent coefficients $a_0[\rho]$, $a_x[\rho]$, and $a_c[\rho]$. *Reprinted from X. Zheng, et al., A generalized exchange-correlation functional: the Neural-Networks approach, Chem. Phys. Lett. 390 (1–3) (2004) 186–192, Copyright (2004), with permission from Elsevier.*

The derivatives are taken on both the coefficients and the individual energy functionals. When one assumes that $a_0[\rho]$, $a_X[\rho]$ and $a_C[\rho]$ do not depend on ρ too much:

$$\frac{\delta a_0[\rho]}{\delta\rho(\mathbf{r})} \approx \frac{\delta a_X[\rho]}{\delta\rho(\mathbf{r})} \approx \frac{\delta a_C[\rho]}{\delta\rho(\mathbf{r})} \approx 0, \tag{4}$$

the second part of the expression vanishes, and the XC potential becomes

$$v_{XC}(\mathbf{r}) \approx a_0[\rho]\frac{\delta E_C^{\text{Slater}}}{\delta\rho(\mathbf{r})} + (1-a_0[\rho])\frac{\delta E_X^{\text{HF}}}{\delta\rho(\mathbf{r})} + a_X[\rho]\frac{\delta \Delta E_X^{\text{Becke}}}{\delta\rho(\mathbf{r})} + a_C[\rho]\frac{\delta E_C^{\text{LYP}}}{\delta\rho(\mathbf{r})} + (1-a_C[\rho])\frac{\delta E_C^{\text{VWN}}}{\delta\rho(\mathbf{r})}. \tag{5}$$

With the above approximation, the machine-learned XC potential was trained and tested in the KS SCF calculations for 116 small molecules, and improvements were observed over the original B3LYP functional (Table 1).

Despite the relative convenience of implementation, there is one key difficulty: the transferability problem. Large amounts of data on small molecules can be readily obtained employing high-precision quantum chemistry methods such as coupled cluster singles and doubles (CCSD) [52] and quantum Monte Carlo (QMC) [53,54]; however, data on large molecules are hard to obtain. How to transfer the knowledge learnt from small molecules to larger molecules is a challenge. Consider when generalizing the functional from small molecules to large molecules: the density descriptors need to be carefully designed and validated; otherwise, different parts of the neural

TABLE 1 Performance of the ML global density functional.

RMS errors (all data are in the units of kcal mol^{-1})					
Properties	**AE**	**IP**	**PA**	**TAE**	**Overall**
Number of samples	56	42	8	10	116
A[a]	2.9	3.9	1.9	4.1	3.4
DFT-1[b]	3.0	4.9	1.6	10.3	4.7
DFT-NN[c]	2.4	3.7	1.6	2.7	2.9

[a] *Becke's work.*
[b] *Conventional B3LYP/6–311 + G(3df,2p).*
[c] *Neural-Networks-based B3LYP/6–311 + G(3df,2p).*
Reprinted from X. Zheng, et al., A generalized exchange-correlation functional: the Neural-Networks approach, Chem. Phys. Lett. 390 (1–3) (2004) 186–192, Copyright (2004), with permission from Elsevier.

network may scale incorrectly as the system sizes grow. Moreover, a descriptor that works for small systems may not be able to capture information as effectively in a large system as different types of physics may contribute differently to systems of different sizes.

For a size-consistent ML DFT model, a more natural choice would be build on the concept of locality. A machine-learned density functional that only depends on local densities can be easily extendable from small to large systems. However, does the local density contain enough information for local XC potential? As will be demonstrated in the following paragraphs, instead of the local electron density, the 'quasi-local electron density' includes enough information for the corresponding local XC potential and therefore all other information about the system. It is worth mentioning that for a local/quasi-local ML model, one may also start from the above-mentioned formulation. Eq. (2) can be generalized into a local/quasi-local one by simply converting to an $\boldsymbol{r}$-dependent version:

$$E_{XC} = \int a_0(\mathtt{r}) e_X^{\text{Slater}}(\boldsymbol{r}) + (1 - a_0(\mathtt{r})) e_X^{\text{HF}}(\mathtt{r}) + a_X(\mathtt{r}) \Delta e_X^{\text{Becke}}(\mathtt{r}) + a_C(\mathtt{r}) e_C^{\text{LYP}}(\mathtt{r}) + (1 - a_C(\mathtt{r})) e_C^{\text{VWN}}(\mathtt{r}) \mathrm{d}^3\mathtt{r}, \tag{6}$$

where the lowercase e_X^{Slater}, e_X^{HF}, $\Delta e_X^{\text{Becke}}$, e_C^{LYP}, and e_C^{VWN} are the local contributions toward the energies of the functionals E_X^{Slater}, E_X^{HF}, $\Delta E_X^{\text{Becke}}$, E_C^{LYP}, and E_C^{VWN}, respectively. As we move forward, it will become clear that this formulation constitutes a special case of the general quasi-local electron density formulation.

Quasi-local electron density formulation of ML-DFTXC: The ML XC potential model

The holographic electron density theorem (HEDT) and its implications on ML-DFTXC

As early as in 1981, Riess and Münch [55] have stated that an arbitrary finite volume of the ground-state electron density determines the density distribution of a molecular system. The

conjecture originates from their hypothesis that the electron density functions in atomic and molecular species are real analytic in the real space excluding the nuclei. The validity of such a hypothesis is, however, neither trivial nor obvious and was yet to be proven rigorously. In 1999, Mezey extended Riess and Münch's statement to the ground-state holographic electron density theorem (GS-HEDT) [56]; however, his proof was not rigorous. Later, Fournais et al. laid a solid mathematical foundation of the GS-HEDT by proving the real analyticity of electron density function of arbitrary atomic and molecular eigenstates [57,58]. Another proof of real analyticity of electron density has been given by Jecko [59]. The GS-HEDT is believed to be related to the concept of quantum similarity measures in DFT [60,61].

In mathematics, a real function is said to be analytic if it possesses derivatives of all orders at every point and agrees with its Taylor series in a neighborhood of each individual point. For any real system such as one consisting of atoms and molecules, the external potential acting on each electron, $\nu(\boldsymbol{r})$, is real analytic except at the nuclei. Apart from the isolated points where the point charges of the nuclei lead to non-analytic electron densities, the r-space is continuous elsewhere and the electron density on it, $\rho(\boldsymbol{r})$, is thus real analytic. In practice, analytic functions such as Gaussian functions and plane waves are often adopted as basis sets for quantum mechanical calculations, which naturally results in real analytic electron densities. As a result, the real analytic $\rho(\boldsymbol{r})$ within a subregion is sufficient to settle its values everywhere in the entire physical space. This can be proven by the analytic continuation of real analytic functions, and an explicit proof can be found in the textbook [62]. A simple proof for the holographic property of real analytic $\rho(\boldsymbol{r})$ in three-dimensional physical space has been provided by Chen et al. [63]. They have also proposed the time-dependent holographic electron density theorem (TD-HEDT) for open electronic systems and applied it to the study of the time-dependent quantum transport problem [63–66].

As is indicated by the GS-HEDT, in principle, the electron density function within any finite volume determines the global density distribution of a real atomic or molecular system. Therefore, DFT can, in principle, yield accurate predictions with the complete information on the electron density function at a single spatial point (including all orders of density derivatives). Nevertheless, in practice, it is more convenient to attain a numerically feasible KS mapping $\rho(\boldsymbol{r}) \rightarrow v_{\mathrm{XC}}(\boldsymbol{r})$, with the electron density in a finite region surrounding the given spatial point $\boldsymbol{r}$, termed as the quasi-local electron density at $\boldsymbol{r}$.

The essence of GS-HEDT has been utilized implicitly in the development of ML-DFTXC. A CNN-based KS mapping has been established, which yields the XC potential at any spatial point by the quasi-local electron density function around that point [3], as shown by Eq. (7). Here, "quasi-local" means that the information on electron density is not rigorously restricted to the interested point $\boldsymbol{r}$ but allowed to include a small neighborhood surrounding that point. Given the impressive accuracy of the test results in reference [3], the compact region selected to represent the quasi-local electron density is considered to be adequate for the determination of the XC potential.

Kohn has proposed the following locality principle on the "nearsightedness of electronic matter" with the following remarks: "local electronic properties, such as the density $\rho(\boldsymbol{r})$, depend significantly on the effective external potential only at nearby points" [67,68]. In fact, the GS-HEDT shares the same foundation as the nearsightedness principle, both suggesting the local nature of ground-state electron density. Although many modern density functional

approximations (DFAs), especially the ones above the third rung of Jacob's ladder [69], intensively adopt the exact exchange or other non-local information for more accurate predictions, it should be possible to achieve a reasonable quasi-local KS mapping by making use of sophisticated ML techniques. Intuitively, size consistency naturally follows (quasi-)locality. In addition, in a quasi-local model, the computational complexity only grows linearly with respect to the system size.

$$v_{\mathrm{XC}}[\rho](\mathfrak{r}) = v_{\mathrm{XC}}[\rho(\mathfrak{r}'||\mathfrak{r}-\mathfrak{r}'|<\delta,\forall\delta)](\mathfrak{r}). \tag{7}$$

To build an ML-DFTXC model of quasi-local electron density, one may start with an XC potential model, where the ML model directly outputs the XC potential to be used in KS SCF calculation. The XC potential plays the central role in DFT [51]. It is predicted by the ML algorithm through quasi-local electron density.

$$v_{\mathrm{ML-XC}}(\mathfrak{r}) = \mathrm{M}_\theta\left[\rho(\mathfrak{r}')\big|_{\mathfrak{r}'\in\mathrm{B}(\mathfrak{r})}\right] \tag{8}$$

$$\left[-\frac{\hbar}{2m}\nabla^2 + v_{\mathrm{ext.}}(\mathfrak{r}) + v_{\mathrm{H}}(\mathfrak{r}) + v_{\mathrm{ML-XC}}\left[\rho(\mathfrak{r}')\big|_{\mathfrak{r}'\in\mathrm{B}(\mathfrak{r})}\right]\right]\psi_i = \varepsilon_i\psi_i \tag{9}$$

where M_θ denotes the ML model with all its parameters θ already optimized, B($\boldsymbol{r}$) denotes the quasi-local neighborhood of $\boldsymbol{r}$, square brackets of $v_{\mathrm{ML\text{-}XC}}$ denote the functional dependency (dependence on the whole neighborhood), and parentheses denote a function. $v_{\mathrm{ext.}}$, v_{H}, and $v_{\mathrm{ML\text{-}XC}}$ refer to the external potential, the Hartree potential, and the machine-learned XC potential, respectively. As dictated by the HEDT, the neighborhood could be arbitrarily small in principle. In practice, the neighborhood must be of a certain size, such that the density values sampled in it can encode the density information in a numerically meaningful way.

For each value of XC potential at grid point $\boldsymbol{r}$, the inputs are sampled from the grid points $\boldsymbol{r}'$ in B($\boldsymbol{r}$) with $\boldsymbol{r}$ at the center, making the model only aware of the relative position $\boldsymbol{r}'-\boldsymbol{r}$. For easier convolution operation, the neighborhood can be chosen as a cube centered at position $\boldsymbol{r}$, with sampling points arranged along their x, y, and z positions in order [3]. For a given window half-length h, the sampling points $\boldsymbol{r}'$ are in the ranges $r'_x\in[r_x-h,r_x+h]$, $r'_y\in[r_y-h,r_y+h]$, and $r'_z\in[r_z-h,r_z+h]$, with certain step length (the smaller the step, the more points sampled given a fixed h). The output is the local XC potential value at $\boldsymbol{r}$, and therefore, once trained, the model predicts the XC potential at the position $\boldsymbol{r}$ of the center of the sampling box.

For different $\boldsymbol{r}$, the XC potential $v_{\mathrm{XC}}(\boldsymbol{r})$ is predicted point-by-point using the same structure and the same set of optimized parameters; therefore, the entire potential is acquired by sweeping the model across the entire grid. The resulting potential is then fed back into the KS equations to calculate a new density.

Pre-calculating XC potential as the target

The electron densities that are to be employed to train the ML model can be obtained using the highly accurate ab initio methods such as the wave-function-based methods such as CCSD. Besides the electron density ρ, v_{XC} is also needed. Given a density from CCSD, the corresponding XC potential v_{XC} can be calculated by various optimization procedures that effectively invert the KS equations (collectively referred to as the inverse Kohn–Sham

methods [70]). The optimization procedures employed in reference [3] to generate training targets is the Wu–Yang (WY) method [71], which will be elaborated here.

Readers might be wondering that if a numerical optimization procedure can solve an XC potential from a density, then why do we bother training an ML model that also predicts an XC potential from a density. The answer lies in the core concept of DFT itself. What we want to machine-learn is the universal XC functional that maps any density to it corresponding XC potential. On the other hand, the optimization procedure only solves a system-specific XC potential that corresponds to a particular known electron density. The procedure entails only the mathematics of the (inverse of) KS equations which does not include the physics about the many-particle system at all. The ML model, on the other hand, tries to learn the actual physics inside, which is by definition universal. The inverse KS method generates the XC potentials v_{XC} which are used, together with electron densities ρ, as the training data fed to the model.

The solution to the inverse KS problem is not as straightforward as it first appears. An analytical solution is mostly absent, and different kinds of numerical optimization procedures are usually employed. One of the popular potential optimization schemes was invented by Wu and Yang [71,72]. For a given input density ρ_{in}, one first constructs a Lagrange W_{s} in terms of the total effective potential v and the single-particle wave functions (WFs) ϕ.

$$W_{\mathrm{s}}[\psi, v(\mathbf{r})] = 2\sum_{i}^{\mathrm{occ.}}\langle\phi_i|\widehat{T}\Big|\phi_j\Big\rangle + \int v(\mathbf{r})\{\rho(\mathbf{r}) - \rho_{\mathrm{in}}(\mathbf{r})\}dr \tag{10}$$

where v plays the role as the Lagrange multiplier. When W_{s} is stationary with respect to v, the electron density becomes the same as the given density ρ_{in}.

$$\frac{\delta W_{\mathrm{s}}[\Psi[v(\mathbf{r})], v(\mathbf{r})]}{\delta v(\mathbf{r})} = \rho(\mathbf{r}) - \rho_{\mathrm{in}}(\mathbf{r}) = 0 \tag{11}$$

In practice, the potential is projected onto a set of Gaussian basis functions. Once the effective potential is calculated, the XC potential v_{XC} can be easily found by subtracting the external potential and the Hartree potential [73].

With the pair of density and XC potential being pre-calculated, the training procedure is decoupled from the KS procedure, and the resulting ML model maps its inputs ρ to the outputs v_{XC}. Training proceeds with a typical backpropagation procedure such as stochastic gradient decent (SGD) [74] or Adam [75]). Once enough data are provided for various types of molecules and various quasi-local environments, the parameters in the ML XC potential model can be optimized to learn and yield the XC potentials v_{XC} of real molecular systems.

Model building, training, and SCF explained with a successful story

The ML model was successfully developed and implemented in 2019 [3]. Densities and potentials were discretized on a grid whose points coincide with the set of quadrature points used for potential integration. For the input, on each integration quadrature point, because neighboring quadrature points usually do not conform to the sampling kernel shape, an extra layer of grid was nested, whose $9 \times 9 \times 9$ cube mesh resamples density and its derivatives in the quasi-local neighborhood. A CNN model was applied on each cube of sampled density, and the data went through a pipeline of two convolution layers, one max-pooling layer, and four fully connected layers, consecutively. The final output of the model is a scalar value of the XC potential at the respective quadrature point. A constant reference potential is usually

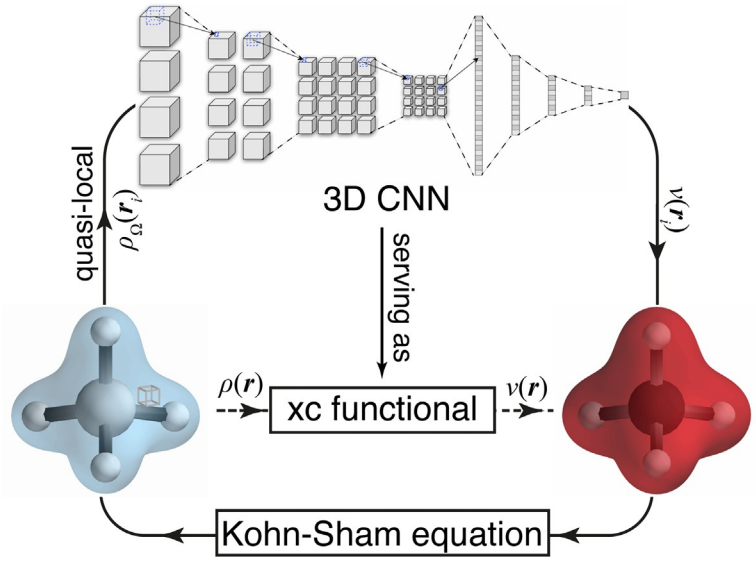

FIG. 2 The structure of the 3-D CNN model for molecules discretized on grid points. The input data at each grid point include density and its derivatives along x, y, and z directions sampled on a $9 \times 9 \times 9$ grid in the respective neighborhood. The output is a scalar value representing the local XC potential. The model includes two convolutional layers and four fully connected layers with a max-pooling operation sandwiched in between. The trained model is used as the XC functional in the SCF calculation. *Adapted with permission from Y. Zhou, et al., Toward the exact exchange–correlation potential: a three-dimensional convolutional neural network construct, J. Phys. Chem. Lett. 10(22) (2019) 7264–7269. Copyright (2019) American Chemical Society.*

subtracted from the overall single-particle potential for numerical convenience. In reference [3], the reference potential is the HF [48] potential from converged HF electron density. For each molecule, during each iteration of the KS SCF procedure, the model sweeps across all quadrature points, and the resultant potential is integrated on grid by the saved quadrature weights to produce the v_{XC} in the matrix form (in the corresponding basis set) (Fig. 2).

The model was trained on a database of 50 H_2 molecules and 50 HeH^+ ions with different bond lengths ranging from 0.504 to 0.896 Å (equally distributed). The trained model was applied both on H_2/HeH^+ with other bond lengths and tested on another ion H-He-He-H^{2+}. Compared to traditional functionals such as B3LYP [46], the results are typically 1 order of magnitude more accurate compared to the reference CCSD electron density. The robustness of the trained model was also demonstrated by starting the SCF at different initial densities while converging to the same density.

The SCF procedure is depicted in Fig. 3. The process is similar to a typical SCF procedure in traditional DFT, except with the XC potential being replaced with the ML XC potential model's output, whereas the input is based on the electron density of current iteration.

Impressive performances (Fig. 4) on the converged electron density (compared to the CCSD results) were observed. Compared to B3LYP results, the accuracy improvements can be as large as 1–2 orders of magnitude. In addition, when forces and energies are calculated from the predicted densities, the accuracies are also much better than those of B3LYP.

More importantly, the adoption of quasi-local electron density as input, together with the special design of the deep learning structure, leads to astonishing transferability. In Fig. 5, the

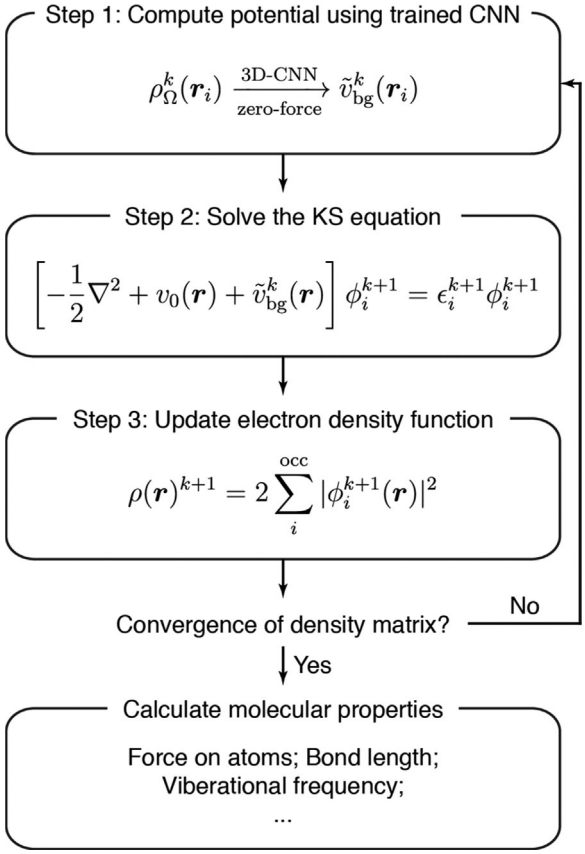

FIG. 3 The SCF process for performing calculations with an ML XC Potential model (the KS-DFT/NN model of the original paper), where k represents the iteration number, v_{bg} represents the XC potential after removing a constant reference part and after projecting onto a basis set, and Ω represents the grid of the 3D box.

same model as shown in previous figures is tested on the HeH^+ ion with much larger He—H distances than that in the training set. Despite being trained on H_2 and HeH^+ of bond distances <0.9 Å, the out-of-sample performance in terms of the density difference to CCSD stayed much smaller than that of B3LYP [46] even at bond distances of $\sim 3\,\mathring{A}$ for HeH^+. More impressively, even when tested in systems such as He-H-H-He^{2+}, which has both different numbers of electrons and different numbers of nuclei than molecules/ions in the training set, the density performance remains better than B3LYP results.

Quasi-local Electron density formulation of ML-DFTXC: The ML XC energy density model

Theory

An alternative target for the ML model is the XC energy density, $\varepsilon_{\text{XC}}(\boldsymbol{r})$ [4], whose unique mapping to the quasi-local density is also guaranteed by the HEDT. The XC energy density $\varepsilon_{\text{XC}}(\boldsymbol{r})$ connects to the XC potential through integration and functional derivatives (to be

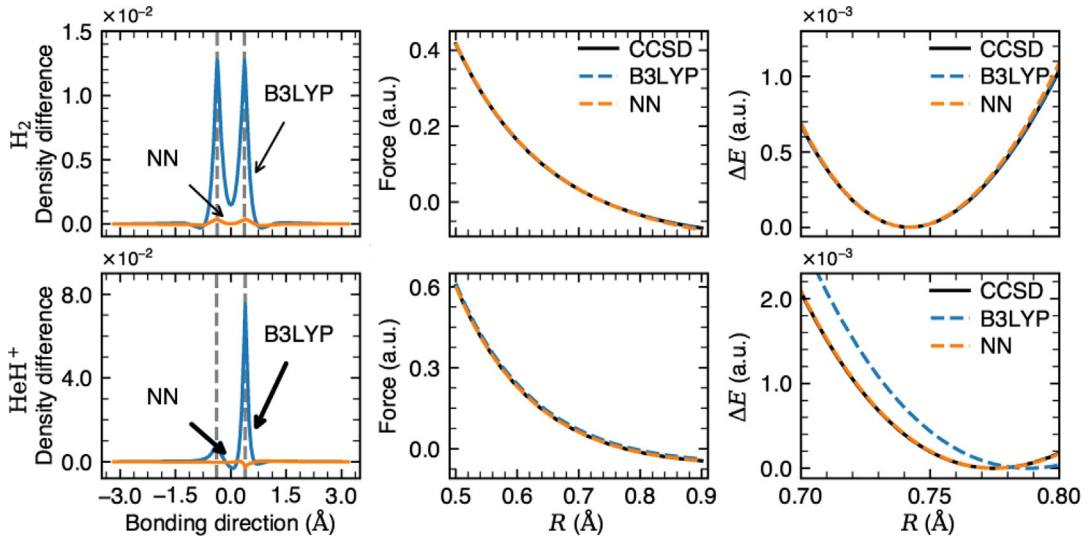

FIG. 4 Performance of the ML XC Potential model (the KS-DFT/NN model of the original paper) in Fig. 2 on H_2/HeH^+. When compared with B3LYP results [46], the density accuracy is 1–2 orders of magnitudes higher. Not surprisingly, the corresponding forces and relative energies calculated based on the densities are also significantly more accurate. *Adapted with permission from Y. Zhou, et al., Toward the exact exchange–correlation potential: a three-dimensional convolutional neural network construct, J. Phys. Chem. Lett. 10(22) (2019) 7264–7269. Copyright {2019} American Chemical Society.*

elaborated in latter sections where we attempt to unify all formalisms). The ML model produces outputs $\varepsilon_{XC}(\boldsymbol{r})$ at each position, which is then swept across all grid points. The energy derivative in the XC potential is calculated at all grid points, and the rest of the procedure is the same as in the ML XC potential model.

By integrating $\varepsilon_{XC}(\boldsymbol{r})$ weighted by $\rho(\boldsymbol{r})$, the ML XC energy density model also output the total XC energy. Given availability of total XC energy data from various sources, it might be tempting to train the model with the total energy as the target alone. However, the complexity of the XC functional entails huge numbers of degrees of freedom. To train a model for an XC functional, targeting only one or a few *scalars* per molecule provides too little information, leaving much room for overtraining. Recall that in the previous section, for each molecule, the model is provided as many data as the number of grid points, either in the form of $v_{XC}(\boldsymbol{r})$ or directly in $\rho(\boldsymbol{r})$. Therefore, the best strategy would be to include training targets on the whole grid. Except at this time, the target can no longer be pre-calculated because there is no procedure such as the WY method [71] to produce energy density $\varepsilon_{XC}(\boldsymbol{r})$. To complicate things even more, because calculating v_{XC} from ε_{XC} (see Eq. (15)) already includes derivatives w.r.t $\rho(\boldsymbol{r})$, calculating the parameters introduces second-order derivates into the computation. In the actual implementation, the second-order derivative calculation includes saving the first derivative graph and other numerical burdens into the backpropagation process; however, with automatic differentiation techniques and packages becoming available, such burdens are no longer deal-breakers.

Coming back to the problem of backpropagation through inverse KS, to circumvent it, one may opt to use optimization procedures that do not rely on backpropagation, for example, a

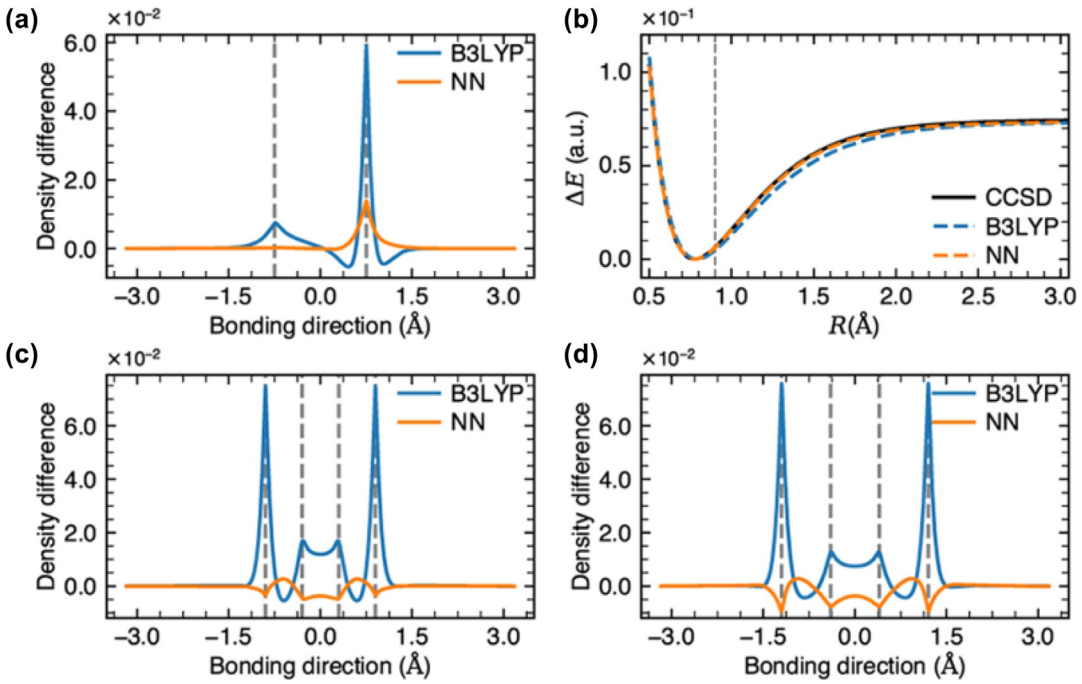

FIG. 5 The out-of-sample performance of the ML XC potential model (KS-DFT/NN model of the original paper). The training set of the model only included the HeH^+ ion up to bond length <0.9 Å. In (a), the density (after SCF) difference (relative to CCSD) of the model on a HeH^+ ion of bond length = 1.5 Å is compared with that of B3LYP [46]. The relative energy of the HeH^+ ion of bond lengths up to 3.0 Å is compared with those of CCSD and B3LYP in (b). The density difference (as in a)) of the linear H_3^+ ion with H—H distance = 0.7 Å is plotted in (c), and the density difference of the linear He-H-H-He^{2+} ion with both H—He and H—H distances = 0.7 Å is plotted in (d). *Adapted with permission from Y. Zhou, et al., Toward the exact exchange–correlation potential: a three-dimensional convolutional neural network construct, J. Phys. Chem. Lett. 10(22) (2019) 7264–7269. Copyright (2019) American Chemical Society.*

Monte Carlo [53] simulation. Optimization without backpropagation is usually much more computationally intensive; however, certain successful cases [4] demonstrated their feasibility when both the model and the dataset are not too large.

Alternatively, one can program the whole KS solver in a fully differentiable way, and backpropagation can be performed together with other differentiable parts in the sequence. Thanks to recent progress in automatic differentiation [76], the differentiable eigen-solver became available, and such kind of backpropagation has been implemented in 1D model systems that were solved directly in real space [5].

Implementation and illustrative examples

In the first work [4], a fully connected neural network was trained with various density descriptors as inputs and the XC energy density as the output. The density descriptors include all or various combinations of the following five quantities:

$$\begin{cases} \rho(\mathbf{r}); \\ \varsigma(\mathbf{r}) = \left(\rho_{\downarrow}(\mathbf{r}) - \rho_{\uparrow}(\mathbf{r})\right)/\rho(\mathbf{r}); \\ s(\mathbf{r}) = |\nabla\rho(\mathbf{r})|/\left[2\left(3\pi^2\right)^{1/3}\rho^{4/3}(\mathbf{r})\right]; \\ \tau(\mathbf{r}) = 1/2\sum_i^{\text{occ.}} |\nabla\varphi_i(\mathbf{r})|^2; \\ R(\mathbf{r}) = \int \mathrm{d}\mathbf{r}'\rho(\mathbf{r}')e^{(-|\mathbf{r}-\mathbf{r}'|/\sigma)}. \end{cases} \tag{12}$$

Depending on which quantities are included, the formulation includes various levels of details about the local/quasi-local density. The ML model becomes a coarse-grained quasi-local electron density model if the fifth term is included (in the original paper [4], when all five descriptors are included; the models are referred to as 'near region approximation' or NRA type functional). When calculating XC potential with Eq. (15), backpropagation was used to calculate the partial derivatives of XC energy with respect to density descriptors. However, in training, backpropagation was replaced with a Monte Carlo [53] method, avoiding the complications from both the 'backpropagation through inverse KS' problem and the second-order derivative problem.

It can be seen from Fig. 6 that for an ML XC energy density model that uses local density descriptors only, the performance is already reasonable. However, the performance only becomes comparable to traditional hybrid functionals when the coarse-grained quasi-local density is included, through the fifth descriptor (the NRA in Fig. 7 as well as in the original paper).

In the second work [5], the KS solver is programmed in a fully differentiable way; in fact, error even backpropagates through multiple iterations of SCF. Because the SCF runs through various densities that oscillate around the converged density, the training effectively includes more information about the functional mapping from the density to the XC energy. Therefore, in the original paper, the scheme was coined the name 'KS regularizer', expressing its generalization (preventing overfitting) capability. The input is the global electron density, and the output is the whole XC energy density. The loss function included both energy error terms and density error terms. The energy error term has contributions from multiple iterations (with a decay factor for earlier iterations), whereas the density error term only contains the root-mean-square error of the last iteration's output density (Fig. 8).

Impressive generalizability was demonstrated for a one-dimensional H_2 model system, where only two training examples were able to determine the whole dissociation curve reasonably well (Fig. 9). However, being developed for 1-D model systems, the work still more or less belongs to the 'proof of concept' category. Considering the computational complexity, an extension toward realistic 3D systems would take extra effort in future. Moreover, such differentiation posts certain numerical issues in degenerate cases. Another promising solution is to change the form of the loss function. Currently, work along this direction is also in progress.

Quasi-local electron density formulation of ML-DFTXC: The ML XC fragment energy model

Theory

As discussed above, the HEDT theorem guarantees the quasi-local density representability of the XC potential as well as the XC energy/energy density. Practically, the information

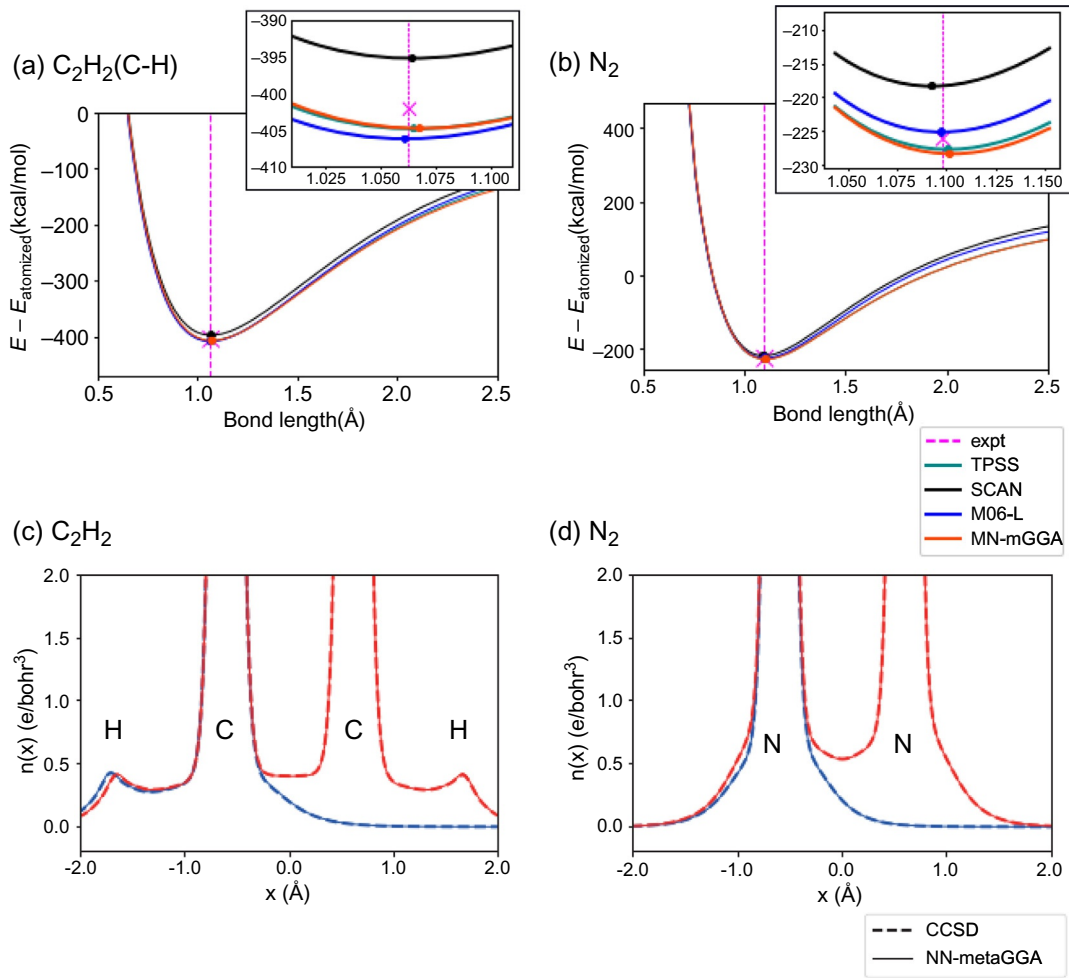

FIG. 6 The performance of an ML XC energy density method developed by Nagai et al. [4] (the 'NN-based meta-GGA' method in the original paper [4]) that uses **local** density information. In (a) and (b), the ML model-predicted dissociation curves for C_2H_2 and N_2 molecules are benchmarked against that of other DFT methods as well as experiments (pink dashed lines and pink crosses as experimental bond distances and experimental atomization energies, respectively). In (c) and (d), the densities from model predictions are compared against CCSD results (for both C_2H_2 and N_2 molecules and the CH radical and the N atom). *Reproduced from reference R. Nagai, R. Akashi, O. Sugino, Completing density functional theory by machine learning hidden messages from molecules, Npj Comput. Mater. 6(1) (2020) 43 under the Creative Commons Attribution 4.0 International license: http://creativecommons.org/licenses/by/4.0/.*

contained in the one-to-one mapping between the local XC potential and quasi-local electron density can be utilized in different ways. One way that differs slightly from previous models is dividing XC energy into contributions from naturally meaningful parts (for example 'atoms'). In Fig. 10, the electron density of a given system is divided into four fragments. By HEDT, there is a unique mapping from each fragmental density to all properties of the

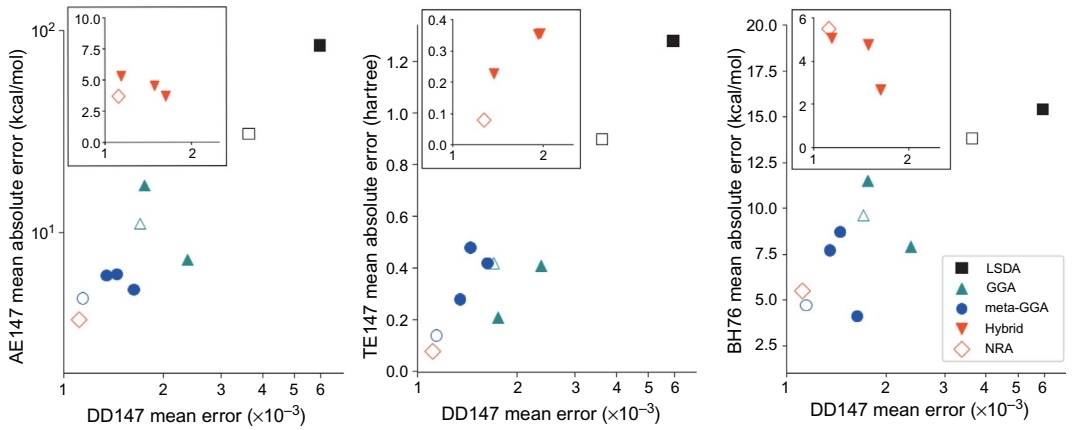

FIG. 7 The performance of an ML XC energy density method developed by Nagai et al. [4] that uses **quasi-local** density information (the NRA in the original paper) benchmarked against various traditional functionals. The local density approximation (LDA) type is represented by the SVWN [48,50] functional; the generalized gradient approximation (GGA) type includes BLYP [46,47] and PBE [77] functionals; meta-GGA includes TPSS [78], SCAN [79], and M06-L [80]; the hybrid includes PBE0 [81], B3LYP [82], and M06 [83], respectively. 'DD147' means that the 'Density Distribution' (root-mean-square error for the density per electron) is benchmarked, and the dataset includes 147 molecules; AE147, TE147, and BH147 mean atomization energy, total energy, and chemical barrier height, respectively. *Reproduced from reference R. Nagai, R. Akashi, O. Sugino, Completing density functional theory by machine learning hidden messages from molecules, Npj Comput. Mater. 6(1) (2020) 43 under the Creative Commons Attribution 4.0 International license: http://creativecommons.org/licenses/by/4.0/.*

system. If a unique way of partitioning the fragmental XC energy contributions $E_{XC} = \sum_i E_i$ is specified, then we have the unique mapping $\rho_{\text{frag}.i} \to \{E_1, E_2, E_3, E_4\} \forall i$. In practice, the mapping to the fragment's own XC energy $\rho_{\text{frag}.i} \to E_i$ would be more straightforward to find, which in turn uniquely determines a quasi-local XC functional $E_i = E_{XC}[\rho_{\text{frag}.\,i}]$. Provided that the way of partitioning a system is universal, this fractional XC functional would also be universal. With atomic division, both the input and output are sampled with reference to atoms. The straightforward way goes as directly equating the total XC energy of a molecule to the summation of XC energy contributions from constituent atoms. An ML model reads and deciphers quasi-local densities around each nucleus, outputting the corresponding atomic XC energy contribution. A derivative of the total XC energy with respect to the electron density yields the XC potential by Eq. (15). It should be noticed that although the XC energy come from an atom-by-atom summation, information from higher-order interactions among atoms can still be encoded because the quasi-local density around each nucleus already includes information from all orders. However, it is up to the ML model to decide the ratio of splitting such energy contribution to the participating parties. For example, for a C═O bond with a specific surrounding, the XC energy correction because of the bonding alone can be assigned to both the carbon and the oxygen and possibly also to other surrounding atoms in small portions.

In fact, molecular potential energy surfaces can be constructed from atomic contributions [44] before the sudden popularity of deep learning models. However, a true XC functional

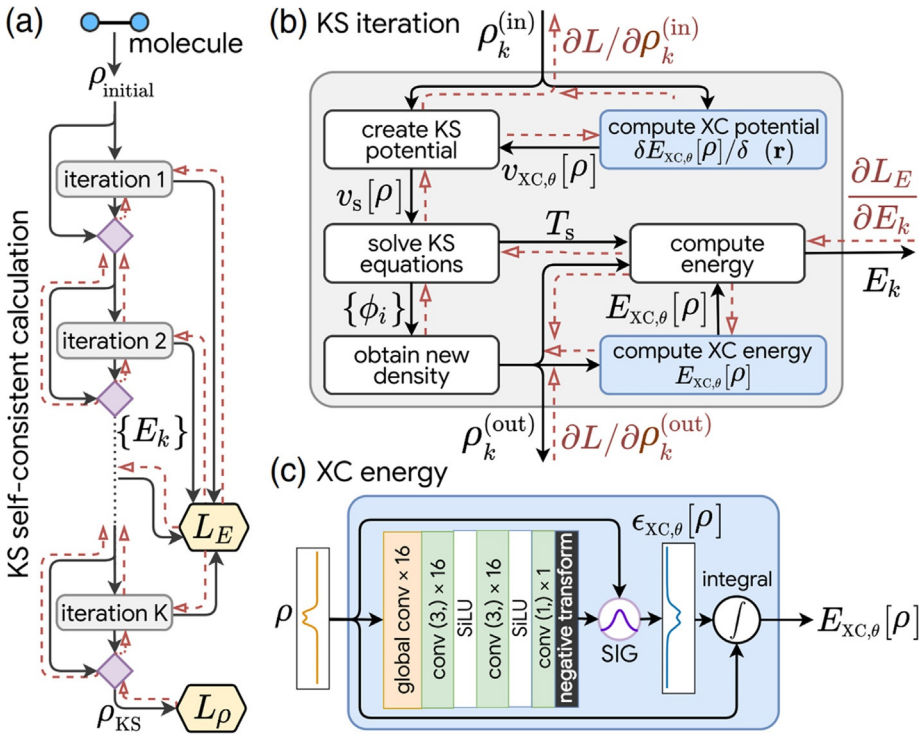

FIG. 8 The structure of the ML XC energy model by Li et al. [5] that includes KS SCF into training. The forward and backward propagation passes in SCF during training are depicted in (a) as *black solid* and read *dashed lines*, respectively, (b) shows the details of one iteration when zoomed in. The structure that utilizes the quasi-local information of the density to produce the XC energy density is depicted in (c). *Adapted from reference L. Li, et al., Kohn-Sham equations as Regularizer: building prior knowledge into machine-learned physics, Phys. Rev. Lett. 126(3) (2021) 036401, with the symbol for density 'n' replaced with 'ρ' for consistency with the rest of the chapter, under the Creative Commons Attribution 4.0 International license: http://creativecommons.org/licenses/by/4.0/.*

should be universal, which needs no extra information about the system other than the density itself, where every nuance of the XC energy or XC potential comes from the subtle differences in the shape of the quasi-local density. To construct a universal XC functional, the complexity of the physics demands models with higher complexity (Fig. 11), which is only becoming practical with recent developments in deep learning.

Implementation and illustrative examples

One successful case of XC energy from atomic contribution demonstrated promising accuracy in small molecules [6]. The model builds specific neural networks for each atom type, sampling the electron density surrounding each nucleus with Gaussian-orbital-like 'projectors'. The descriptors are calculated from integrating the quasi-local electron density with different projectors. The 'projected' values are then symmetrized to become the input for

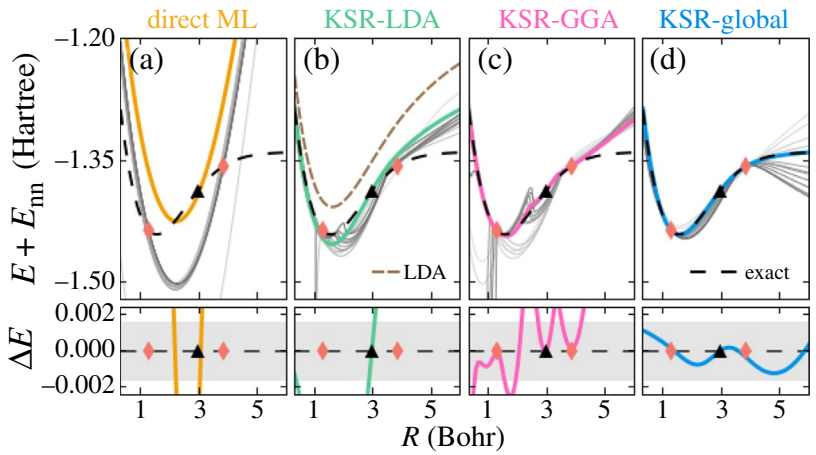

FIG. 9 The performance of the 'KS Regularizer (KSR)' model using either the local density information (the KSR-LDA and the KSR-GGA) or the quasi-local (the KSR-global in the graph) density information versus no KSR (direct ML). The models are trained using data from two structures (*red* (*gray* in print version) *diamonds*), and parameters were further selected (validation) by performances on one structure (*black triangle*). Gray curves represent the performance of models during training iterations. *Reproduced from reference L. Li, et al., Kohn-Sham equations as Regularizer: building prior knowledge into machine-learned physics, Phys. Rev. Lett. 126(3) (2021) 036401 under the Creative Commons Attribution 4.0 International license: http://creativecommons.org/licenses/by/4.0/.*

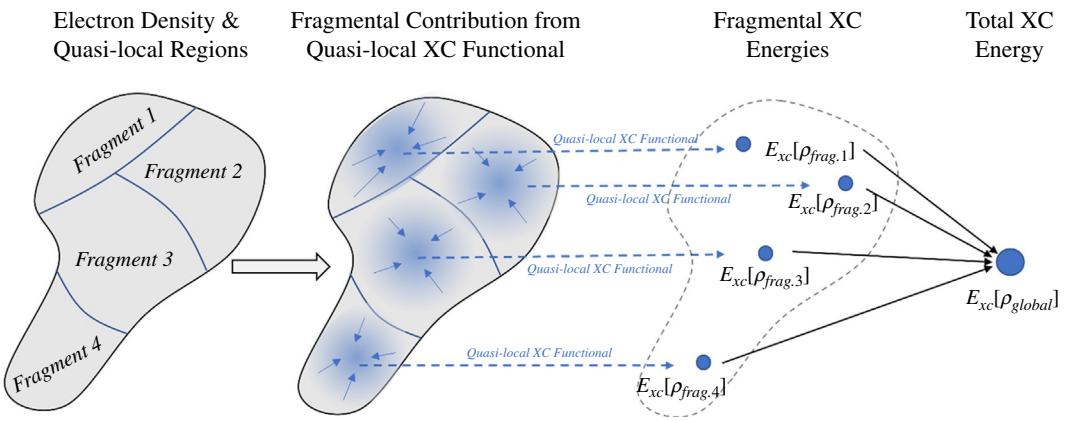

FIG. 10 The concept of the XC Energy by Fragments. In practice, the fragments are usually chemically meaningful parts like electron density around each nucleus (atom) in a molecule.

the neural networks. The output of each neural network is a scalar value representing energy contribution from each atom. The total XC energy is calculated from summing the outputs of all atomic neural networks (Fig. 11). For SCF calculation (Fig. 12), functional derivatives need to be taken with respect to density. Noticeably, the derivatives assume rather simple transformation from density descriptors to density itself:

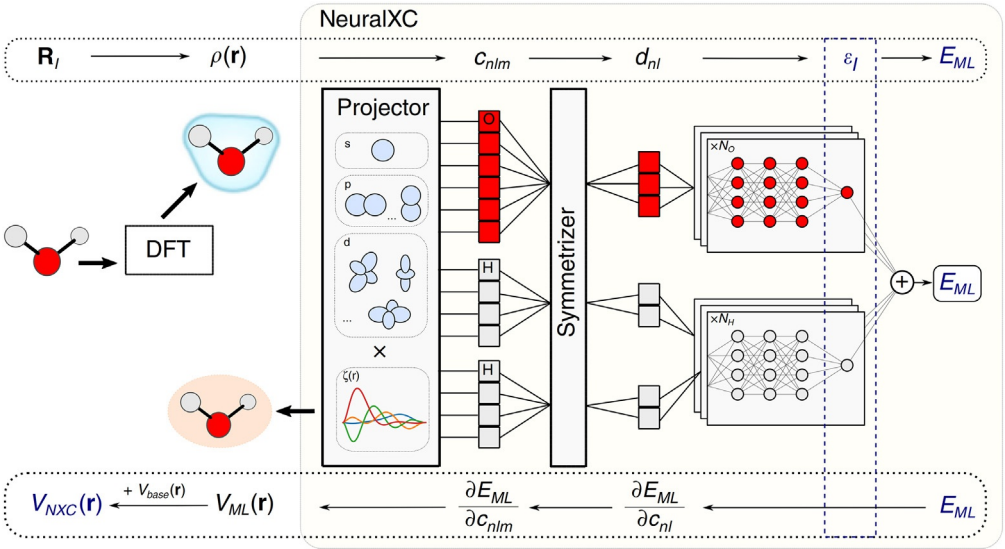

FIG. 11 The structure of the ML XC energy by atomic contribution. The quasi-local electron density around each nucleus is described by projecting on to 'projectors'. Descriptors are then symmetrized and fed to atomic neural networks. Atoms of the same type share the same neural network parameters. Their respective outputs play the role as fragmental contributions, which are summed up to produce the total XC energy. *Reproduced from reference S. Dick, M. Fernandez-Serra, Machine learning accurate exchange and correlation functionals of the electronic density, Nat. Commun., 11(1) (2020) 1–10 under the Creative Commons Attribution 4.0 International license: http://creativecommons.org/licenses/by/4.0/.*

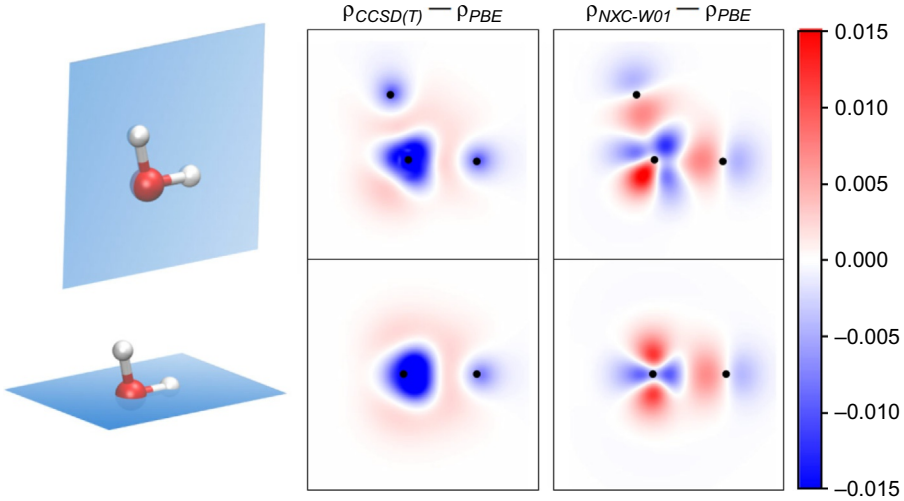

FIG. 12 The density differences between the ML XC energy model (NXC-001 in the original paper)-predicted electron density and PBE-calculated electron density for water molecules versus that between CCSD(T) and PBE. *Reproduced from reference S. Dick, M. Fernandez-Serra, Machine learning accurate exchange and correlation functionals of the electronic density, Nat. Commun., 11(1) (2020) 1–10 under the Creative Commons Attribution 4.0 International license: http://creativecommons.org/licenses/by/4.0/.*

$$v_{\mathrm{ML}}[\rho(\mathtt{r})] = \sum_{\beta} \frac{\partial E_{\mathrm{ML}}}{\partial c_\beta} \frac{\delta c_\beta[\rho]}{\delta \rho(\mathtt{r})} = \sum_{\beta} \frac{\partial E_{\mathrm{ML}}}{\partial c_\beta} \psi_\beta(\mathtt{r}), \tag{13}$$

where β is the index for different projectors, c_β is the projected value of the density on the projector, and $\psi(\boldsymbol{r})$ is the shape of the projector.

However, the model(s) is not yet universal because not only different atoms use different networks and also different projectors but also, more importantly, different models were trained for different datasets (three different models were trained for three different datasets in reference [6]).

General quasi-local electron density formalism of ML-DFTXC

In all the three types of quasi-local models introduced earlier, the ML model connects the election density to the XC energy, XC energy density, or XC potential. In this section, we unify all of them while trying to elucidate deeper insights among their connections. The input of the ML algorithm is the quasi-local density, whereas the output can be any of the three: the XC potential, the XC energy density, or the fragmental/atomic XC energy contribution. The XC potential relates to the XC energy through partial derivatives with respect to the electron density, and the XC energy density relates to the XC energy through an integration:

$$E_{\mathrm{XC}} = \int \varepsilon_{\mathrm{XC}}(\mathtt{r})\rho(\mathtt{r})d\mathtt{r} = \int \varepsilon_{\mathrm{XC}}[\rho(\mathtt{r}'|\mathtt{r}'\in \mathrm{B}(\mathtt{r})), \mathtt{r}]\rho(\mathtt{r})\mathrm{d}\mathtt{r}, \tag{14}$$

$$\begin{aligned} v_{\mathrm{XC}}(\mathtt{r}) &= \frac{\delta E_{\mathrm{XC}}}{\delta \rho(\mathtt{r})} = \varepsilon_{\mathrm{XC}}(\mathtt{r}) + \int_{\mathtt{r}'\in \mathrm{B}(\mathtt{r})} \frac{\delta \varepsilon_{\mathrm{XC}}(\mathtt{r}')}{\delta \rho(\mathtt{r})} \rho(\mathtt{r}')\mathrm{d}\mathtt{r}' \\ &= \varepsilon_{\mathrm{XC}}\left[\rho(\mathtt{r}'|\mathtt{r}'\in B(\mathtt{r})), \mathtt{r}\right] + \int_{\mathtt{r}'\in \mathrm{B}(\mathtt{r})} \frac{\delta \varepsilon_{\mathrm{XC}}[\rho(\mathtt{r}''|\mathtt{r}''\in \mathrm{B}(\mathtt{r}').), \mathtt{r}']}{\delta \rho(\mathtt{r})} \rho(\mathtt{r}')\mathrm{d}\mathtt{r}'. \end{aligned} \tag{15}$$

Here again, B(r) denotes the neighborhood of any point $\boldsymbol{r}$ in space. With the electron density given, all the three quantities are uniquely determined, and the conversion from one to another is also uniquely specified. It is worth reiterating that despite being formulated in XC potential, the HEDT works for both XC potential (v_{XC}) and XC energy density ($\varepsilon_{\mathrm{XC}}$) at every point $\boldsymbol{r}$ in space; therefore, an ML model's structural design for XC energy density can be directly borrowed from that for the XC potential and vice versa.

For the ML XC fragment energy model, the density-weighted integration in (15) is replaced with a summation over fragmental contribution. For a smooth transition from fragment to fragment, fragmental boundaries are usually not clear-cut (for example in (13) the 'projector' c_β has a kernel of Gaussian orbital shape [6]); therefore, at any position $\boldsymbol{r}$, multiple E_is can contribute to $v_{\mathrm{XC}}(\boldsymbol{r})$:

$$v_{\mathrm{XC}}(\mathtt{r}) = \frac{\delta E_{\mathrm{XC}}}{\delta \rho(\mathtt{r})} = \sum_{i:r\in \mathrm{Frag}.i} \frac{\delta E_i}{\delta \rho(\mathtt{r})}. \tag{16}$$

Eventually, the machine-learned model is plugged into KS SCF calculations, where the XC potential is needed to solve the KS equations. Therefore, the total energy and energy density

approaches need extra conversion from the ML models outputs. Once the ML XC potential is calculated, it is added on top of other terms that do not depend on the ML model, including the external potential, the Hartree potential, and an optional custom reference potential. A new electron density is calculated from the solution of the KS equations. The new density is then fed into the same ML model, and the same procedure is repeated. Therefore, like in traditional DFT, the system is solved iteratively until convergence. Once the SCF converges, the final density will then be used to calculate a variety of molecular properties such as the electron affinity, the ionization potential, the heat of formation and so on.

It should be noted that because the ML model represents the universal XC *functional* rather than a single system-specific XC potential, the same ML model with the same *parameters* is applicable to all atoms, molecules, ions, and materials.

However, during training, the parameter optimization is complicated by the iterative procedure of SCF for the ML XC energy density model and the ML XC fragment energy model. The ML model and the KS equations are tangled together in the above-mentioned formulations. Intuitively, to train the ML model, the parameters should be optimized in such a way that helps the SCF procedure converge to the correct density. To implement this, the algorithm should minimize the density error everywhere in space. However, optimizing the whole SCF procedure is complicated and implicit. Theoretically, one may present the formalism as an SCF procedure nested in a model optimization procedure; practically, for realistic systems, a simpler view is called for.

Because the same model is invoked during each iteration, one may isolate one iteration out from the SCF procedure and focus optimization/training only on it. Considering the fact that when SCF reaches convergence, the density settled down around the target density. Therefore, the model should be optimized in such a way that the correct density gives back the correct density itself after one iteration. To summarize, the optimization is simplified to 'right density in, right density out, one iteration in between'.

To implement such an optimization procedure, a loss function can be written directly in terms of the input density and the newly calculated density from one SCF iteration,

$$L_\rho = \mathrm{E}_{\text{training samples}}\left[\int \left(\rho^{\text{ML+KS}}(\mathtt{r}) - \rho^{\text{target}}(\mathtt{r})\right)^2 \mathtt{dr}\right], \tag{17}$$

where $\rho^{\text{ML+KS}}$ is generated from ρ^{target} by running one SCF iteration and $\mathrm{E}_{\text{training samples}}$ indicates taking expectation over all training samples (molecules in the training set). To train the model by backpropagating a loss function of this form, one could implement the process of solving KS equations in a fully differentiable way. One may also choose other forms of loss functions in terms of density that make backpropagation through KS easier by certain mathematical manipulations.

On the other hand, if the XC potential is the ML model's output, one may skip the KS equations altogether during training (but only for training) and write the loss function directly in terms of the XC potential itself as

$$L_{v_{\text{XC}}} = \mathrm{E}_{\text{training samples}}\left[\int \left(v_{\text{XC}}^{\text{ML}}(\mathtt{r}) - v_{\text{XC}}^{\text{target}}(\mathtt{r})\right)^2 \mathtt{dr}\right]. \tag{18}$$

In this case, the target v_{XC} should be *pre-calculated* in the data preparation phase, and the model is algorithmically decoupled from the KS equation altogether during training.

On the other hand, if the model output is chosen to be the XC energy $E_{XC}[\rho_{quasi-local}(\boldsymbol{r})]$ or the XC energy density $\varepsilon_{XC}(\boldsymbol{r})[\rho_{quasi-local}(\boldsymbol{r})]$, in addition to reproducing the density, the model will output the energy through Eq. (14), for which an extra scalar loss (19) can be added. The energy loss function can be combined with the density loss function with a weight factor in front, which is then tuned as a hyperparameter.

$$L_E = \mathrm{E}_{\text{training samples}}\left[E_{XC}^{ML} - E_{XC}^{Target}\right]^2. \tag{19}$$

In the training phase, all parameters in the ML model will be optimized to reproduce the density either directly or indirectly (and to reproduce the energy and/or energy derived properties if included). The target ground-state density is usually acquired through expensive ab initio methods such as CCSD or CCSD(T), whereas additional energy-related molecular properties are either from ab initio methods to or from experiments. Optimization in training is usually performed by descending along the loss function's gradient direction. The gradient (expressed in terms of the model parameters) is most effectively calculated through backpropagation. If the density loss function (17) is included and the model is coupled with KS equations, as mentioned above, backpropagation needs to go through the inverse eigenvalue problem in the KS equations first before reaching the ML model. It takes certain numerical techniques or special engineering of the loss functions for the backpropagation to access the network where the actual parameters are updated. Alternatively, reproducing the target density can be enforced by using Eq. (18) only, which leads back to pre-calculating XC potential as the target and the successful story in 2.3.

Additional ML models: ML For van der Waals interaction

Van der Waals (vdW) interaction plays a central role in organic chemistry, biochemistry, and material sciences. it is related to transient dipoles, producing little manifestation in the electron density. The subtlety of such interaction plaques both ML-DFT and traditional DFT alike.

Although many conventional DFAs with relatively simple forms (such as LDA, GGAs, and hybrids) fail to describe the vdW accurately, several other schemes exhibit remarkable performance in certain systems [84]. However, most successful stories rely heavily on non-local quantities, hindering the direct transplantation to the quasi-local ML-DFT framework.

Given the fact that vdW is mainly because of the interactions among transient dipole moments and barely induces density change, a combined ML model is called for. Indeed, transient dipoles' interaction does not come from changes in the respective interacting densities. The very minor density response is the result, rather than the reason, for the vdW interaction. The minor density change and its corresponding v_{XC} change are both higher in order in a perturbative sense. On the other hand, the energy shift because of vdW interaction is a direct effect of the transient dipoles' interaction.

To elaborate, it is clear from (20) that when calculating v_{XC} during SCF, the effect from the minor density change of vdW interaction can be largely ignored within reasonable accuracy requirements; indeed, the second term with second-order derivatives with respect to ρ is much smaller than the first term as $\delta\rho << 1$ for vdW interaction. On the other hand, the energy shift is obviously not negligible. Therefore, adding an additional correction term for

vdW after SCF becomes the most sensible choice. In fact, a vdW ML model can be trained separately based on quasi-local electron density. After training, it can be added on top of the ML XC models as an extra correction term to the XC energy.

$$\begin{aligned} E_{\mathrm{XC}}[\rho_0+\delta\rho] &\approx E_{\mathrm{XC}}[\rho_0]+\int \left.\frac{\delta E_{\mathrm{XC}}[\rho]}{\delta\rho(\mathbf{r})}\right|_{\rho=\rho_0}\delta\rho(\mathbf{r})\mathrm{d}\mathbf{r}, v_{\mathrm{XC}}[\rho_0+\delta\rho,\mathbf{r}]=\left.\frac{\delta E_{\mathrm{XC}}[\rho]}{\delta\rho(\mathbf{r})}\right|_{\rho=\rho_0+\delta\rho} \\ &\approx \left.\frac{\delta E_{\mathrm{XC}}[\rho]}{\delta\rho(\mathbf{r})}\right|_{\rho=\rho_0}+\int_{\mathbf{r}'\in B(\mathbf{r})}\left.\frac{\delta^2 E_{\mathrm{XC}}[\rho]}{\delta\rho(\mathbf{r})\delta\rho(\mathbf{r}')}\right|_{\rho=\rho_0}\delta\rho(\mathbf{r}')\mathrm{d}\mathbf{r}'. \end{aligned} \tag{20}$$

The ML model for vdW can borrow the conventional wisdom from traditional DFT. In fact, empirical correction approaches such as the DFT-D3 method [85] have been widely used for prediction of geometric structures and energetic properties of dispersion. These approaches are computationally very efficient, but their effectiveness relies heavily on the few empirical parameters and depends critically on the system. On the other hand, a specially designed ML model is perfect for a large number of tunable parameters and degrees of freedom. Provided with enough training data, an ML model for vdW corrections will almost surely outperform currently available fitted models. Once trained and validated, a combination of such an ML model with the quasi-local ML-DFT model would be straightforward.

Case study

An example for the ML-DFTXC potential model

For a hands-on experience, the readers are encouraged to try out our open-source models and datasets on GitHub. Most of our code was written in python, and our models were built with the open-source package PyTorch [86]. For a better understanding of the implementation details, inexperienced readers are recommended to go through a comprehensive tutorial of PyTorch before making any modifications to the models we provide. As a starting point, PyTorch provides introductory level tutorials on their own website at https://pytorch.org/tutorials/.

Our model uses WY XC potential as the direct training target. For the H_2 molecule example (Fig. 13), training can be performed with our pre-calculated CCSD density and WY v_{xc} for a H_2 molecule with a 0.7 $\mathring{A}$ bond length. No SCF calculations are needed for training. At the evaluation phase, full SCF calculation can be performed for the example structure. The SCF calculation is implemented with the PySCF [87] package. To perform training and evaluation on the example, one may follow steps (1) to (7) listed below:

(1) Before getting started, please make sure all the prerequisites are installed and work properly.

(2) Create and enter a new folder, and download our code and dataset by typing
≫*git clone https://github.com/zhouyyc6782/oep-wy-xcnn.git*

(3) Enter the *example/simple_H2* directory, create a folder /log here to store the upcoming results, and run training by typing
≫*python ../nn_train/main.py train.cfg.*

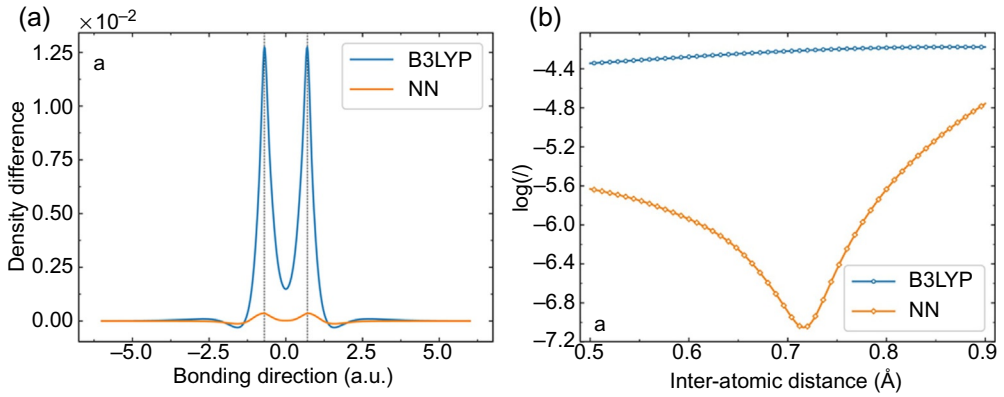

FIG. 13 A typical (with a reasonable bond distance) SCF run for the trained model of the H_2 example will produce density errors comparable to NN in (a), and the I value will be comparable to the corresponding value on the NN curve in (b), which should be significantly lower than the error of B3LYP. *Panel (a) is adapted with permission from Y. Zhou, et al., Toward the exact exchange–correlation potential: a three-dimensional convolutional neural network construct, J. Phys. Chem. Lett. 10(22) (2019) 7264–7269. Copyright (2019) American Chemical Society.*

Here, all training settings and hyperparameters are defined in the .cfg file; to write a new .cfg file for a different configuration, please refer to the README file provided with the code.

(4) Training will start on the provided H_2 dataset; by default, the maximum epochs is 1000.

(5) Perform SCF calculations on the newly trained model by typing

≫*python ../xcnn/main.py test.cfg*

(6) One can check the SCF performance of the model by examining the output file generated. A typical run for a small molecule such as H_2 should result in an error at the level of $10^{-5}\sim10^{-7}$ in terms of *I*. Because only one H_2 structure is included in the *simple_H2* example training set, the error could be larger. Please refer to the original paper for the definition of *I* [3].

(7) The tutorial is centered around a pre-built dataset from one H_2 structure for both training and SCF. To use the whole dataset and reproduce the result from the original publication [3], more v_{xc} target data need to be generated by the WY method. To use the code in the */oep-wy* folder for performing WY calculations, a modified and recompiled version of PySCF is needed. For a set of H_2 and HeH^+ data, the *run_oep.py*, *gen_dataset.py*, *run_train.py* and *run_xcnn.py* provides automatic scripts for generating data from WY calculations, collecting data, training the model with the data, and testing the model with SCF procedures, respectively. Interested readers are advised to follow the README from the GitHub repository in step 2) for recompiling PySCF and additional custom implementations of the code.

The WY, training, and SCF codes we provide here are not limited to the above example or the molecules/ions demonstrated in reference [3]. The readers are encouraged to use the code on different systems. Depending on the format of the dataset, the readers need to write their own scripts similar to *run_oep.py*, *gen_dataset.py*, *run_train.py*, and *run_xcnn.py* mentioned in step 7) for automating the WY, training, and SCF procedures.

Conclusions and outlook

The explosive development in AI catalyzed a quick turnover of deep learning model structure designs. From the algorithm perspective, most of the above-mentioned approaches focused on applying CNN on learning DFT XC functionals, whereas GNN, RNN, and Transformers are also promising candidates for overhauling DFT design. GNN extends CNN toward irregular grids for electron density/XC potential. RNN [88] is ideal for time-dependent data and may find profound applications in TD-DFT. However, Transformers [11] and other attention-based models allow the model to be smarter, deciding by itself where to 'pay attention to' in the electron density or XC potential. The subtlety and sensitivity of the electron density data in DFT problems made it a perfect target for such attention-based models.

During the preparation of this chapter, a new work based on quasi-local electron density formulation of ML-DFTXC was published [89]. This is a quasi-local version of the global electron density formulation of ML-DFTXC reported in reference [2]. Instead of learning the mapping $\rho_{\text{quasi-local}} \rightarrow v_{\text{XC}}/E_{\text{XC}}/\varepsilon_{\text{XC}}$ from scratch, the model learns the $\boldsymbol{r}$-dependent coefficients of three existing functionals.

$$E_{\text{XC}}^{\text{MLP}}[\rho] = \int \mathbf{f}_\theta(x(\mathbf{r})) \cdot \begin{bmatrix} e_{\text{X}}^{\text{LDA}}(\mathbf{r}) \\ e^{\text{HF}}(\mathbf{r}) \\ e^{\omega\text{HF}}(\mathbf{r}) \end{bmatrix} \mathrm{d}^3\mathbf{r}. \tag{21}$$

In Eq. (21), f_θ is a row vector of three elements outputted by the ML model, and the $e_{\text{X}}^{\text{LDA}}(\boldsymbol{r})$, $e^{\text{HF}}(\boldsymbol{r})$, and $e^{\omega\text{HF}}(\boldsymbol{r})$ are the local LDA [48], local Hartree Fock [90], and local range-separated Hartree Fock [90] energies, respectively. An extra D3 [85] correction was added to the $E_{\text{XC}}^{\text{MLP}}$ to produce the final XC energy prediction.

In the chapter, we reviewed several approaches that redesign DFT XC energy functional by ML. We started from a functional model that uses the global density, moving toward more intuitive and transferable quasi-local models, and ended with additional ML term for vdW than can be added on top of the ML XC terms that induce larger electron density changes than those of vdW interaction. For quasi-local models, we brought up the holographic electron density theorem as the theoretical foundation, followed by a series of successful implementation schemes. All schemes of quasi-local ML-DFTXC: the ML XC potential model, the ML XC energy density model, and the ML XC fragment energy model, have deep physical connections and share the same fundamental design elements in common. Successful stories for these variants are demonstrated [3–6], and the readers are encouraged to find more inspiration from the respective original papers, as well as the open-source code and examples we provided. It is our hope that the universal XC functional of DFT can be found and accurately reproduced by new generations of ML models in near future, revolutionizing the field of computational chemistry like what Alphafold [91] did to the field of structural biology.

References

[1] D.J. Tozer, V.E. Ingamells, N.C. Handy, Exchange-correlation potentials, J. Chem. Phys. 105 (20) (1996) 9200–9213.

[2] X. Zheng, et al., A generalized exchange-correlation functional: the neural-networks approach, Chem. Phys. Lett. 390 (1–3) (2004) 186–192.

[3] Y. Zhou, et al., Toward the exact exchange–correlation potential: a three-dimensional convolutional neural network construct, J. Phys. Chem. Lett. 10 (22) (2019) 7264–7269.
[4] R. Nagai, R. Akashi, O. Sugino, Completing density functional theory by machine learning hidden messages from molecules, Npj Comput. Mater. 6 (1) (2020) 43.
[5] L. Li, et al., Kohn-Sham equations as Regularizer: building prior knowledge into machine-learned physics, Phys. Rev. Lett. 126 (3) (2021), 036401.
[6] S. Dick, M. Fernandez-Serra, Machine learning accurate exchange and correlation functionals of the electronic density, Nat. Commun. 11 (1) (2020) 1–10.
[7] A.W. Senior, et al., Improved protein structure prediction using potentials from deep learning, Nature 577 (7792) (2020) 706–710.
[8] D. Silver, et al., Mastering the game of Go with deep neural networks and tree search, Nature 529 (7587) (2016) 484–489.
[9] Y. LeCun, et al., Backpropagation applied to handwritten zip code recognition, Neural Comput. 1 (4) (1989) 541–551.
[10] M. Gori, G. Monfardini, F. Scarselli, A new model for learning in graph domains, in: Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005, IEEE, 2005.
[11] A. Vaswani, et al., Attention is All You Need, Advances in Neural Information Processing Systems 30, NIPS, 2017.
[12] J.P. Perdew, K. Schmidt, Jacob's ladder of density functional approximations for the exchange-correlation energy, in: AIP Conference Proceedings, American Institute of Physics, 2001.
[13] R. Nagai, et al., Neural-network Kohn-Sham exchange-correlation potential and its out-of-training transferability, J. Chem. Phys. 148 (24) (2018), 241737.
[14] X. Lei, A.J. Medford, Design and analysis of machine learning exchange-correlation functionals via rotationally invariant convolutional descriptors, Phys. Rev. Mater. 3 (6) (2019), 063801.
[15] J. Wang, et al., Improving density functional prediction of molecular thermochemical properties with a machine-learning-corrected generalized gradient approximation, J. Phys. Chem. A 126 (6) (2022) 970–978.
[16] T. Nudejima, et al., Machine-learned electron correlation model based on correlation energy density at complete basis set limit, J. Chem. Phys. 151 (2) (2019), 024104.
[17] Y. Ikabata, et al., Machine-learned electron correlation model based on frozen core approximation, J. Chem. Phys. 153 (18) (2020), 184108.
[18] J.T. Margraf, K. Reuter, Making the coupled cluster correlation energy machine-learnable, J. Phys. Chem. A 122 (30) (2018) 6343–6348.
[19] J.T. Margraf, K. Reuter, Pure non-local machine-learned density functional theory for electron correlation, Nat. Commun. 12 (1) (2021) 1–7.
[20] E. Cuierrier, P.-O. Roy, M. Ernzerhof, Constructing and representing exchange–correlation holes through artificial neural networks, J. Chem. Phys. 155 (17) (2021), 174121.
[21] J. Schmidt, C.L. Benavides-Riveros, M.A. Marques, Machine learning the physical nonlocal exchange–correlation functional of density-functional theory, J. Phys. Chem. Lett. 10 (20) (2019) 6425–6431.
[22] R. Han, M. Rodríguez-Mayorga, S. Luber, A machine learning approach for MP2 correlation energies and its application to organic compounds, J. Chem. Theory Comput. 17 (2) (2021) 777–790.
[23] M.F. Kasim, S.M. Vinko, Learning the exchange-correlation functional from nature with fully differentiable density functional theory, Phys. Rev. Lett. 127 (12) (2021), 126403.
[24] G. Yang, et al., Size-independent neural networks based first-principles method for accurate prediction of heat of formation of fuels, J. Chem. Phys. 148 (24) (2018), 241738.
[25] L. Hu, et al., Combined first-principles calculation and neural-network correction approach for heat of formation, J. Chem. Phys. 119 (22) (2003) 11501–11507.
[26] J. Sun, et al., Alternative approach to chemical accuracy: a neural networks-based first-principles method for heat of formation of molecules made of H, C, N, O, F, S, and Cl, J. Phys. Chem. A 118 (39) (2014) 9120–9131.
[27] J. Wu, X. Xu, The X1 method for accurate and efficient prediction of heats of formation, J. Chem. Phys. 127 (21) (2007), 214105.
[28] H. Li, et al., Improving the accuracy of density-functional theory calculation: the genetic algorithm and neural network approach, J. Chem. Phys. 126 (14) (2007), 144101.
[29] X.-M. Duan, et al., Neural network correction for heats of formation with a larger experimental training set and new descriptors, Chem. Phys. Lett. 410 (1–3) (2005) 125–130.

[30] H. Ji, Y. Jung, A local environment descriptor for machine-learned density functional theory at the generalized gradient approximation level, J. Chem. Phys. 148 (24) (2018), 241742.
[31] K. Ryczko, D.A. Strubbe, I. Tamblyn, Deep learning and density-functional theory, Phys. Rev. A 100 (2) (2019), 022512.
[32] Q. Liu, et al., Improving the performance of long-range-corrected exchange-correlation functional with an embedded neural network, J. Phys. Chem. A 121 (38) (2017) 7273–7281.
[33] M. Bogojeski, et al., Quantum chemical accuracy from density functional approximations via machine learning, Nat. Commun. 11 (1) (2020) 1–11.
[34] M. Welborn, L. Cheng, T.F. Miller III, Transferability in machine learning for electronic structure via the molecular orbital basis, J. Chem. Theory Comput. 14 (9) (2018) 4772–4779.
[35] L. Cheng, et al., A universal density matrix functional from molecular orbital-based machine learning: transferability across organic molecules, J. Chem. Phys. 150 (13) (2019), 131103.
[36] L. Li, et al., Understanding machine-learned density functionals, Int. J. Quantum Chem. 116 (11) (2016) 819–833.
[37] K. Yao, J. Parkhill, Kinetic energy of hydrocarbons as a function of electron density and convolutional neural networks, J. Chem. Theory Comput. 12 (3) (2016) 1139–1147.
[38] J. Seino, et al., Semi-local machine-learned kinetic energy density functional with third-order gradients of electron density, J. Chem. Phys. 148 (24) (2018), 241705.
[39] J. Seino, et al., Semi-local machine-learned kinetic energy density functional demonstrating smooth potential energy curves, Chem. Phys. Lett. 734 (2019), 136732.
[40] P. Golub, S. Manzhos, Kinetic energy densities based on the fourth order gradient expansion: performance in different classes of materials and improvement via machine learning, Phys. Chem. Chem. Phys. 21 (1) (2019) 378–395.
[41] F. Brockherde, et al., Bypassing the Kohn-Sham equations with machine learning, Nat. Commun. 8 (1) (2017) 1–10.
[42] J.C. Snyder, et al., Finding density functionals with machine learning, Phys. Rev. Lett. 108 (25) (2012), 253002.
[43] R. Meyer, M. Weichselbaum, A.W. Hauser, Machine learning approaches toward orbital-free density functional theory: simultaneous training on the kinetic energy density functional and its functional derivative, J. Chem. Theory Comput. 16 (9) (2020) 5685–5694.
[44] J. Behler, M. Parrinello, Generalized neural-network representation of high-dimensional potential-energy surfaces, Phys. Rev. Lett. 98 (14) (2007), 146401.
[45] P. Hohenberg, W. Kohn, Inhomogeneous electron gas, Phys. Rev. 136 (3B) (1964) B864–B871.
[46] C. Lee, W. Yang, R.G. Parr, Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density, Phys. Rev. B 37 (2) (1988) 785–789.
[47] A.D. Becke, Density-functional exchange-energy approximation with correct asymptotic-behavior, Phys. Rev. A 38 (6) (1988) 3098–3100.
[48] J.C. Slater, A simplification of the Hartree-Fock method, Phys. Rev. 81 (3) (1951) 385–390.
[49] V. Fock, Näherungsmethode zur Lösung des quantenmechanischen Mehrkörperproblems, Z. Phys. 61 (1–2) (1930) 126–148.
[50] S.H. Vosko, L. Wilk, M. Nusair, Accurate spin-dependent electron liquid correlation energies for local spin density calculations: a critical analysis, Can. J. Phys. 58 (8) (1980) 1200–1211.
[51] W. Kohn, L.J. Sham, Self-consistent equations including exchange and correlation effects, Phys. Rev. 140 (4A) (1965) A1133–A1138.
[52] J. Čížek, On the correlation problem in atomic and molecular systems. Calculation of wavefunction components in Ursell-type expansion using quantum-field theoretical methods, J. Chem. Phys. 45 (11) (1966) 4256–4266.
[53] N. Metropolis, et al., Equation of state calculations by fast computing machines, J. Chem. Phys. 21 (6) (1953) 1087–1092.
[54] W.L. McMillan, Ground state of liquid He^4, Phys. Rev. 138 (2A) (1965) A442–A451.
[55] J. Riess, W. Münch, The theorem of hohenberg and kohn for subdomains of a quantum system, Theor. Chim. Acta 58 (4) (1981) 295–300.
[56] P.G. Mezey, The holographic electron density theorem and quantum similarity measures, Mol. Phys. 96 (2) (1999) 169–178.
[57] S. Fournais, et al., Analyticity of the density of electronic wavefunctions, Ark. Mat. 42 (1) (2004) 87–106.
[58] S. Fournais, et al., The Electron Density is Smooth Away from the Nuclei, Commun. Math. Phys. 228 (2002) 401–415.

[59] T. Jecko, A new proof of the analyticity of the electronic density of molecules, Lett. Math. Phys. 93 (1) (2010) 73–83.
[60] P. Geerlings, et al., Density functional theory and quantum similarity, Int. J. Quantum Chem. 101 (6) (2005) 722–732.
[61] R. Carbó-Dorca, E. Besalú, Communications on quantum similarity (2): a geometric discussion on holographic electron density theorem and confined quantum similarity measures, J. Comput. Chem. 31 (13) (2010) 2452–2462.
[62] S.G. Krantz, H.R. Parks, A Primer of Real Analytic Functions, Springer Science & Business Media, 2002.
[63] X. Zheng, et al., Time-dependent density-functional theory for open systems, Phys. Rev. B 75 (19) (2007), 195127.
[64] X. Zheng, et al., Existence of time-dependent density-functional theory for open electronic systems: time-dependent holographic electron density theorem, Phys. Chem. Chem. Phys. 13 (32) (2011) 14358–14364.
[65] X. Zheng, et al., Time-dependent density functional theory for quantum transport, J. Chem. Phys. 133 (11) (2010), 114101.
[66] X. Zheng, G. Chen, First-principles method for open electronic systems, in: Z. Tang, P. Sheng (Eds.), Nanoscale Phenomena, Springer, 2008, pp. 235–243.
[67] W. Kohn, Density functional and density matrix method scaling linearly with the number of atoms, Phys. Rev. Lett. 76 (17) (1996) 3168.
[68] E. Prodan, W. Kohn, Nearsightedness of electronic matter, Proc. Natl. Acad. Sci. 102 (33) (2005) 11635–11638.
[69] J.P. Perdew, et al., Prescription for the design and selection of density functional approximations: more constraint satisfaction with fewer fits, J. Chem. Phys. 123 (6) (2005), 062201.
[70] Y. Shi, A. Wasserman, Inverse Kohn–Sham density functional theory: progress and challenges, J. Phys. Chem. Lett. 12 (22) (2021) 5308–5318.
[71] Q. Wu, W. Yang, A direct optimization method for calculating density functionals and exchange–correlation potentials from electron densities, J. Chem. Phys. 118 (6) (2003) 2498–2509.
[72] W. Yang, Q. Wu, Direct method for optimized effective potentials in density-functional theory, Phys. Rev. Lett. 89 (14) (2002), 143002.
[73] D.R. Hartree, The wave mechanics of an atom with a non-Coulomb central field. Part I. Theory and methods, in: Mathematical Proceedings of the Cambridge Philosophical Society, Cambridge University Press, 1928.
[74] Robbins, H. and S. Monro, A stochastic approximation method. Ann. Math. Stat., 1951. 22(3): p. 400–407, 8.
[75] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, arXiv preprint arXiv:1412.6980, arXiv.org, Ithaca, NY, 2014.
[76] A.G. Baydin, et al., Automatic differentiation in machine learning: a survey, J. Mach. Learn. Res. 18 (2018).
[77] J.P. Perdew, K. Burke, M. Ernzerhof, Generalized gradient approximation made simple, Phys. Rev. Lett. 77 (18) (1996) 3865.
[78] J. Tao, et al., Climbing the density functional ladder: nonempirical meta–generalized gradient approximation designed for molecules and solids, Phys. Rev. Lett. 91 (14) (2003), 146401.
[79] J. Sun, A. Ruzsinszky, J.P. Perdew, Strongly constrained and appropriately normed semilocal density functional, Phys. Rev. Lett. 115 (3) (2015), 036402.
[80] Y. Zhao, D.G. Truhlar, A new local density functional for main-group thermochemistry, transition metal bonding, thermochemical kinetics, and noncovalent interactions, J. Chem. Phys. 125 (19) (2006), 194101.
[81] M. Ernzerhof, G.E. Scuseria, Assessment of the Perdew–Burke–Ernzerhof exchange-correlation functional, J. Chem. Phys. 110 (11) (1999) 5029–5036.
[82] P.J. Stephens, et al., Ab initio calculation of vibrational absorption and circular dichroism spectra using density functional force fields, J. Phys. Chem. 98 (45) (1994) 11623–11627.
[83] Y. Zhao, 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. Accounts 120 (1) (2008) 215–241.
[84] N. Tasinato, S. Grimme, Unveiling the non-covalent interactions of molecular homodimers by dispersion-corrected DFT calculations and collision-induced broadening of ro-vibrational transitions: application to (CH2F2)2 and (SO2)2, Phys. Chem. Chem. Phys. 17 (8) (2015) 5659–5669.
[85] S. Grimme, et al., A consistent and accurate ab initio parametrization of density functional dispersion correction (DFT-D) for the 94 elements H-Pu, J. Chem. Phys. 132 (15) (2010), 154104.
[86] A. Paszke, et al., Pytorch: an imperative style, high-performance deep learning library, Adv. Neural Inf. Proces. Syst. 32 (2019) 8026–8037.

[87] Q. Sun, et al., PySCF: the Python-based simulations of chemistry framework, Wiley Interdiscip. Rev. Comput. Mol. Sci. 8 (1) (2018), e1340.
[88] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning Internal Representations by Error Propagation, MIT Press, 1985.
[89] J. Kirkpatrick, et al., Pushing the frontiers of density functionals by solving the fractional electron problem, Science 374 (6573) (2021) 1385–1389.
[90] J. Jaramillo, G.E. Scuseria, M. Ernzerhof, Local hybrid functionals, J. Chem. Phys. 118 (3) (2003) 1068–1073.
[91] J. Jumper, et al., Highly accurate protein structure prediction with AlphaFold, Nature 596 (7873) (2021) 583–589.