www.nature.com/scientificreports

scientific reports

OPEN

Application of two-component neural network for exchange-correlation functional interpolation

Alexander Ryabov[1,2]✉, Iskander Akhatov[1] & Petr Zhilyaev[1]

Density functional theory (DFT) is one of the primary approaches to solving the many-body Schrodinger equation. The essential part of the DFT theory is the exchange-correlation (XC) functional, which can not be obtained in analytical form. Accordingly, the accuracy improvement of the DFT is mainly based on the development of XC functional approximations. Commonly, they are built upon analytic solutions in low- and high-density limits and result from quantum Monte Carlo or post-Hartree-Fock numerical calculations. However, there is no universal functional form to incorporate these data into XC functional. Instead, various parameterizations use heuristic rules to build a specific XC functional. The neural network (NN) approach to interpolate the data from higher precision theories can give a unified path to parametrize an XC functional. Moreover, data from many existing quantum chemical databases could provide the XC functional with improved accuracy. We develop NN XC functional, which gives exchange potential and energy density without direct derivatives of exchange-correlation energy density. Proposed NN architecture consists of two parts NN-E and NN-V, which could be trained in separate ways, adding new flexibility to XC functional. We also show that the developed NN XC functional converges in the self-consistent cycle and gives reasonable energies when applied to atoms, molecules, and crystals.

Since its emergence, density functional theory (DFT)[1,2] serves as one of the primary methods of solving the many-body Schrodinger equation. The main theoretical bottleneck in DFT theory is the unknown form of the exchange-correlation (XC) functional. Therefore, the progress in developing more accurate XC functionals reveals more possibilities for using DFT in cases where high accuracy of quantum-mechanical calculations is required. Information for constructing XC functionals is taken from numerical calculations using quantum Monte Carlo or post-Hartree-Fock[3,4]. The influential Monte Carlo (MC) simulations of the uniform electron gas (UEG) by Ceperley and Adler[3] led to the creation a number of practical local density approximations (LDAs)[5–7]. The next big success in reaching better accuracy was achieved by the generalized gradient approximation (GGA), which takes into consideration local gradients of electron density[8–10]. This improvement sufficiently increased the capability of DFT to characterize systems with inhomogeneous electron densities. Calculations made by post-Hartree-Fock methods were also used to improve the quality of the XC functionals for molecular systems[4,11]. The search for advanced XC functional is ongoing, and, still a very active direction of research[12–15].

Regardless of the evident triumph of the LDA and GGA, their development is a highly complex procedure that involves many heuristics stages. The XC functional's form is generally specified by physical insights (the local nature of the interaction, perturbation approach, analytical solutions in limiting cases, etc.), and a set of adjustable parameters[7,10]. Such inflexible functional form makes it difficult to include more numerical results from modern quantum MC[16,17] and post-Hartree-Fock calculations[18], which could lead to increased XC functional accuracy. Therefore it may be fruitful to use an adaptive XC functional form that, on the one hand, facilitates the incorporation of numerical calculations and, on the other hand, enable to include the physical insights into it.

One perspective candidate for the flexible XC functional form is the neural network (NN), which provides a universal approach to approximate any functional relationship[19]. Analytical information could be also included in the NN as a synthetically generated part of the dataset. NN was first utilized as a functional form for XC potential by Tozer et al.[20]. After that, several studies have been addressed the possibility of using NN to approximate XC

[1]Center for Materials Technologies, Skolkovo Institute of Science and Technology, Bolshoy Boulevard 30, bld. 1, Moscow 121205, Russia. [2]Moscow Institute of Physics and Technology (State University), Institutskiy per. 9, Dolgoprudny, Moscow Region 141700, Russia. ✉email: ununbium17@gmail.com

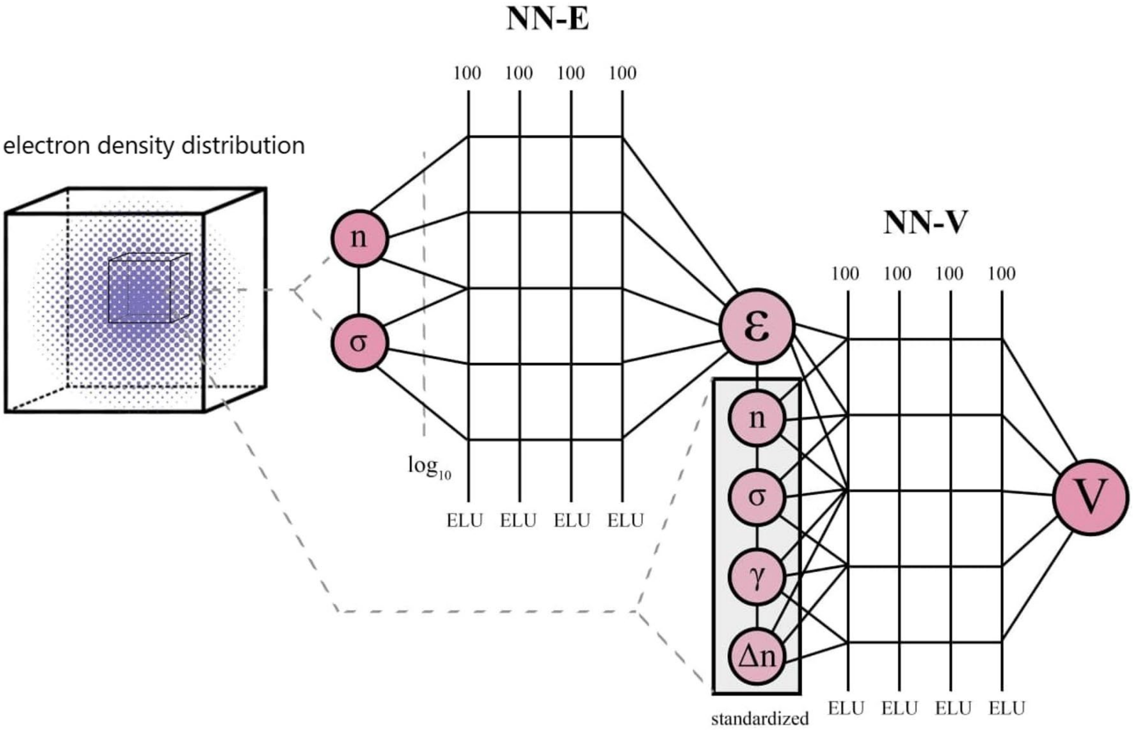

Figure 1. Topology of the XC neural network. It consists of two parts: NN-E predicts ε_{xc} and NN-V predicts v_{xc}. Each part of the neural network consists of 4 layers each of 100 neurons. For both parts information on local density and its derivatives is needed.

functionals form[21–26]. Work by Nagai et al.[22] is especially worthy of note. They presented the first working NN XC functional, which gives better accuracy than traditional ones.

Despite significant advances in the development of XC functionals based on NNs, a wide range of issues remain. Namely, how to relate the exchange-correlation energy density (ε_{xc}) and the exchange-correlation potential (XC potential, V_{xc})? In the analytical description, such connection is provided by a standard differentiation, which is sometimes tedious but straightforward. But when new features are included in the NN, for example, the logarithm of density (standard feature scaling procedure) or local Hartree-Fock exchange energy densities[27], it is not clear how to incorporate them into the connection between XC potential and corresponding energy density. It is also essential to find a way to include physical insight into XC functionals, such as asymptotic analytical solutions and conservation laws. Another issue is an optimal NN architecture and feature selection for presenting XC functional. So far, no detailed comparison has been made of various NN architectures on the same training data.

In this study we focus on constructing NN XC functional that output both ε_{xc} and V_{xc}. The developed NN consists of two parts: one part is used to evaluate ε_{xc} (NN-E), and another approximates V_{xc} (NN-V). They connected in such a way that the output of NN-E is one of the input features for NN-V (see Fig. 1). Proposed architecture is also "economical", since one back-propagation step does not need to traverse the entire mesh, and it is calculated only at one point. In other practical approaches[22,27], for one step of the neural network, it is necessary to calculate the total exchange-correlation energy of the system, which significantly increases memory consumption and therefore leads to limitations for training and evaluation of neural networks with a large number of parameters. Direct differentiation of features to obtain the exchange-correlation functional from the density of the exchange-correlation energy can be quite a complex and confusing problem. In particular, if the features are semi-local or global, obtaining the final analytic expression is non-trivial. An example is the exchange energy density as a feature in the work[27]. Local coordinate transformations only complicate the task.

The training dataset was obtained from the DFT calculation of crystalline silicon, benzene, and ammonia with PBE XC functional[10,28]. After XC NN training, we implemented it into Octopus DFT code[29–31] and conducted self-consistent cycle calculations of train/test systems and atoms and molecules from IP13/03 dataset[32]. The mean relative error of total and XC energies on training samples is on the order of 0.001%. The same errors on crystals and molecules that were not used to train XC NN increased by the order magnitude but were still small, around 0.01%. Such a small relative error indicates that the proposed architecture could be successfully used for XC functional form representation. The key feature of the proposed NN architecture is that the weights of NN-V are pre-trained on $\varepsilon_{xc} \to V_{xc}$ mapping and fixed during learning on new training data. It allows being sure that the output of NN-E is indeed the ε_{xc}, because the relation between ε_{xc} and V_{xc} is preserved.

It is also should be noted that the boundary conditions are included by using extra datasets. They are synthetically generated to fulfill given boundary conditions. In present case it was $\varepsilon_{xc} \to 0$ given that electron density (n) is vanishing, and $\varepsilon_{xc} \to \varepsilon_{xc}^{LDA}$ given that gradient modulus squared of electron density (σ) is also approaching zero.

This paper is outlined as follows: In "Methods" section, we describe the data generation process, NN topology, features and hyperparameters used. In the "Results" section, we first show the results of NN training, features distributions, and evaluated metrics; in the following, we demonstrate NN XC's application in DFT calculations on atoms and molecules that were not presented in the training dataset. In "Conclusion" section, we summarize our results and suggest following research steps that could be done with the developed NN XC pipeline, with a particular emphasis on integrating data from high-level methods such as quantum MC and post-HF calculation.

Methods

The proposed neural network architecture consists of two parts: the NN-E and the NN-V. The NN-E serves to obtain the exchange-correlation energy density (ε_{xc}), and the NN-V allows calculating of the exchange-correlation potential (V_{xc}) from the corresponding energy using the output of NN-E as one of the neurons. Only the spin unpolarized case is considered. Generalization to the spin-polarized case is trivial and can be implemented by increasing the number of input neurons corresponding to the spin components appropriately.

The NN-E input parameters are electron density n and the square of the electron density gradient $\sigma = \langle \nabla n, \nabla n \rangle$. We use base-10 logarithmic transformation for preprocessing input features of NN-E. The input features for the NN-V are ε_{xc}, $n, \sigma, \gamma = \langle \nabla \sigma, \nabla n \rangle$ and Laplacian of the electron density Δn. All NN-V features except ε_{xc} are standardized, i.e. converted to zero mean and unit variance (see Fig. 1). The choice of features for NN-V is due to the functional connection between ε_{xc} and V_{xc} in generalized gradient approximation (GGA), which contains the electron density, its gradients in various combinations and the Laplacian. In a real space grid the connection between potential and XC energy density looks as following[33]:

$$\begin{aligned} v_{\mathrm{xc}}^{\mathrm{GGA}} =& \varepsilon_{\mathrm{xc}}^{\mathrm{GGA}} + n\frac{\partial \varepsilon_{\mathrm{xc}}^{\mathrm{GGA}}}{\partial n} - g\frac{\partial \varepsilon_{\mathrm{xc}}^{\mathrm{GGA}}}{\partial g} - ng\frac{\partial^2 \varepsilon_{\mathrm{xc}}^{\mathrm{GGA}}}{\partial g \partial n} \\ & - \frac{n}{g}\frac{\partial \varepsilon_{\mathrm{xc}}^{\mathrm{GGA}}}{\partial g}\nabla^2 n + \frac{n}{g^2}\frac{\partial \varepsilon_{\mathrm{xc}}^{\mathrm{GGA}}}{\partial g}\vec{g}\nabla g - \frac{n}{g}\frac{\partial^2 \varepsilon_{\mathrm{xc}}^{\mathrm{GGA}}}{\partial g^2}\vec{g}\nabla g, \end{aligned} \tag{1}$$

where $g(\mathbf{r}) \equiv |\nabla n(\mathbf{r})|$ and $\vec{g} \equiv \nabla n(\mathbf{r})$. In a broad sense, the choice of features is arbitrary and the only criterion for its success is the increase in the accuracy of NN.

At the first stage of training, only the NN-V is trained. In this case, the energy that is supplied to the input to the NN-V is obtained using the libXC[34] package. The first stage aims to teach mapping between the ε_{xc} and the corresponding potential, and the ε_{xc} known in advance is used. In this case the following loss function is applied:

$$Loss = \frac{1}{N}\sum_{i=1}^{N}(V_{xc}^{PBE}[i] - V_{xc}^{predicted}[i])^2. \tag{2}$$

At the second stage, the NN-V weights are frozen, and only NN-E part is trained, but loss depends on the output of NN-V (V_{xc}). The second stage aims to train the neural network to map electron density, σ and V_{xc}. Simultaneously, the frozen NN-V provides the correct connection between ε_{xc} and V_{xc}. At this stage we additionally include boundary condition in the loss function:

$$\begin{aligned} Loss = \frac{1}{N}\sum_{i=1}^{N} & [(V_{xc}^{PBE}[i] - V_{xc}^{predicted}[i])^2 \\ & + (\varepsilon_{xc}^{predicted}(0, \sigma)[i] - 0)^2 \\ & + (\varepsilon_{xc}^{predicted}(n, 0)[i] - \varepsilon_{xc}^{LDA}(n)[i])^2]. \end{aligned} \tag{3}$$

The first boundary condition is based on the fact that ε_{xc} tends to zero at zero density with any input σ. The second boundary condition follows from the fact that with zero σ and any density ε_{xc} leads to the corresponding energy of the local density approximation.

To obtain the training data, we carry out DFT calculations[1,2] of silicon, benzene, and ammonia. We perform the calculations in real space using the Octopus code[29–31]. The XC functionals used include the exchange and correlation parts from PBE[10,28]. For all atomic species pseudo-potentials SG15 are used[35]. The total number of the calculations is 10 for each chemical substance. Atomic configurations in these calculations are differed by applied strain ±5%. Grid spacing in the range from zero to 10 is used. This corresponds to $64 \times 64 \times 64$ mesh for silicon and benzene, and $65 \times 65 \times 65$ mesh for ammonia. The mesh size for ammonia is a little bit larger due to technical issues related to non-periodic boundary conditions. For avoiding gradient inconsistency in the boundaries we use cropping. We remove data points lying at a distance of 4 (due to numerical scheme of differentiation) or less from the boundaries of the parallelepiped. Finally, the dataset size was approximately 5.2 million samples.

The Pytorch framework[36] is utilized for training the neural network. We use the Adam algorithm for training with a learning rate descending from 0.001 by 25 percent every 20 epoch, and mean square error (MSE) loss. The batch size selected for training is 5000. The neural architecture used for the NN-E is a fully connected network with two input neurons, one linear output neuron, and four exponential linear units (ELU)[37] hidden layers with 100 neurons. NN-V has the same architecture except that the number of input neurons increased to 5. To implement boundary conditions (3) we also included two additional batches in each training step. The first one consists of zero electron density and non-zero gradients with corresponding zero ε_{xc}. The second one contains non-zero electron density and zero gradients with corresponding ε_{xc}^{LDA}. Thus, on the one hand, we

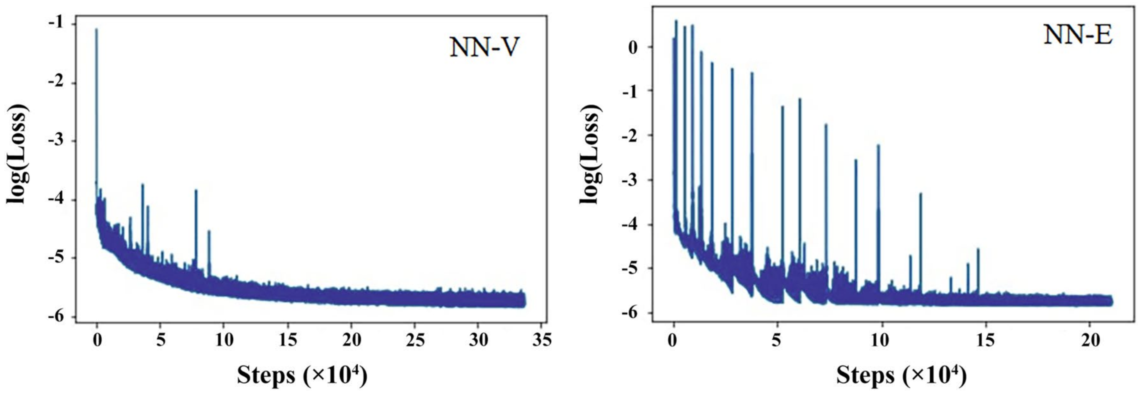

Figure 2. Training curves of XC neural network. The left one corresponds to NN-V training, where log(Loss) is the logarithm of loss (2). The right one corresponds to NN-E training where log(Loss) is the logarithm of loss with boundary conditions (3). Outliers on the training curves are related with adjustable learning rate used for optimization algorithm.

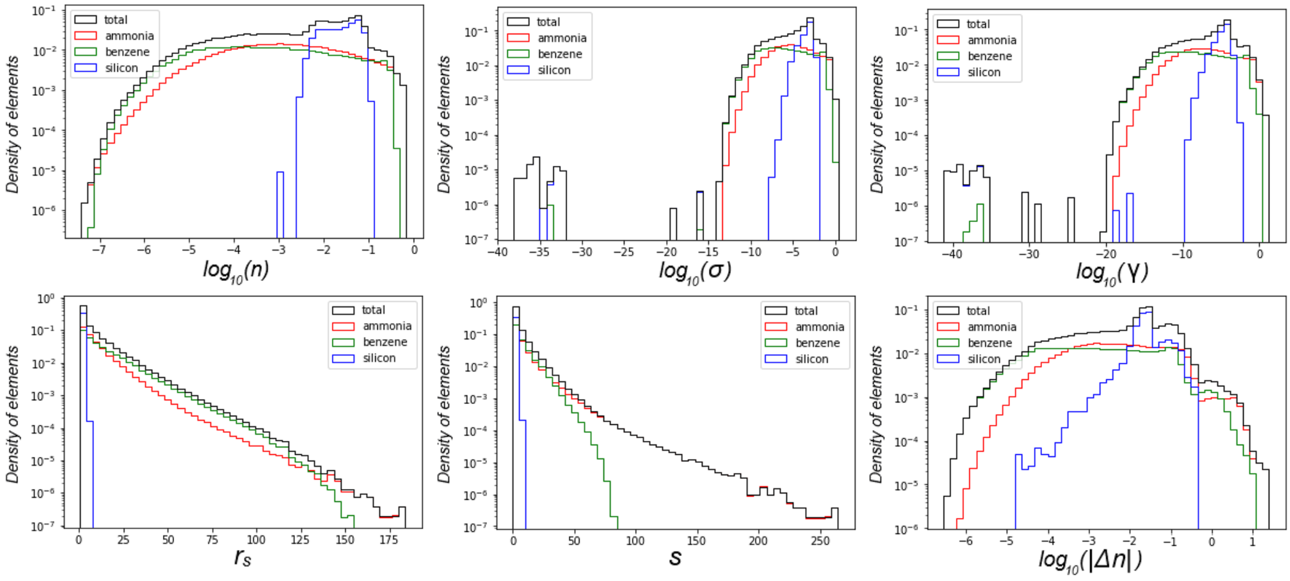

Figure 3. Distributions of input features. Each curve is normalized to to the total number of elements in the corresponding dataset.The distributions localized around the values that are typical for the systems used in training dataset: ammonia, benzene, crystalline silicon. One can see that crystalline silicon has relatively small electron density gradients and no region of low electron density due to its periodic structure. If a neural network is given an input value significantly out of bounds of the distribution data, it potentially could output an inappropriate value.

have included the boundary conditions directly in the loss function; on the other hand, we have reinforced the boundary conditions directly with data.

Results

Training of neural network.

The training of NN was performed on benzene, silicon and ammonia. For each substance ten calculation was performed. In each calculation the system was stretched or compressed in range of ±5% of lattice constant to obtain electron densities in a wider range.

Training curves for NN-V and NN-E are presented in Fig. 2. The loss reaches a plateau after 300k batches for NN-V and 200k batches for NN-E correspondingly. Each batch consisted of 32 sample. The final loss achieved values of about $10^{-6}\,\mathrm{Ha}^2 \times a_0^6$ in both cases.

The distributions of input features for training dataset are presented in Fig. 3. Individual distributions of substances included in the training dataset are normalized to the total number of elements corresponding to the given substance. In additional to features used distribution by the Wigner–Seitz radius $r_s = (3/(4\pi n))^{1/3}$ and the reduced density gradient $\mathrm{s} = |\nabla \mathrm{n}|/2\mathrm{k_F n} = |\nabla n|/(2\sqrt[3]{3}\pi^{2/3}n^{4/3})$ are presented. All distribution is not uniform due to real systems used in training procedure. The distributions localized around the values that are typical for the systems under consideration. Analysis of such distributions is important to determine the limits

	MSE (ε_{xc})	MAE (ε_{xc})	MSE (V_{xc})	MAE (V_{xc})
Benzene	4×10^{-7}	2×10^{-4}	1×10^{-6}	6×10^{-4}
Silicon	2×10^{-5}	1×10^{-3}	1×10^{-6}	4×10^{-4}
Ammonia	2×10^{-6}	4×10^{-4}	3×10^{-6}	1×10^{-3}

Table 1. MSE and MAE results for ε_{xc} and V_{xc} on a training dataset. The units of MSE are $\mathrm{Ha}^2 \times a_0^6$, the units of MAE are $\mathrm{Ha} \times a_0^3$.

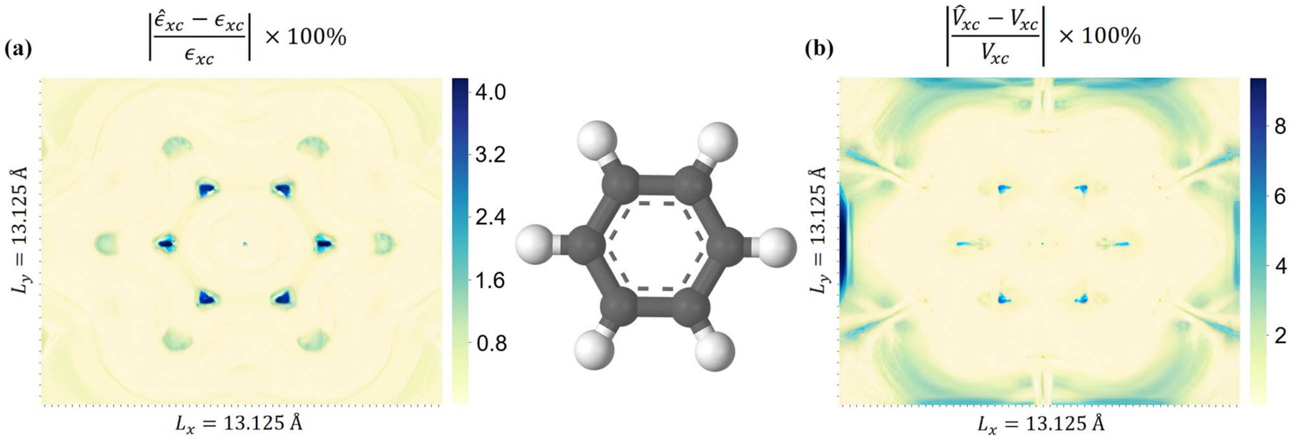

Figure 4. Spacial distribution of relative error for benzene: (**a**) spacial distribution of relative error for ε_{xc}, (**b**) spacial distribution of relative error for V_{xc}. The relative local errors have highest values in the region of the nuclei and at the boundary.

of NN XC functional applicability, i.e. to detect cases where NN XC functional will certainly fail due to absence of such data in the training dataset. This is especially important for case of heavy atoms in which core electron density could have large derivatives.

We use absolute mean square error and (MSE) and mean average error (MAE) as metrics to evaluate learning outcomes:

$$MSE(x) = \frac{1}{N}\sum_{n=1}^{N}\left(\hat{x}_n - x_n\right)^2, \tag{4}$$

$$MAE(x) = \frac{1}{N}\sum_{n=1}^{N}\left|\hat{x}_n - x_n\right|, \tag{5}$$

where x_n is the true value of target variable, $\hat{x}_n$ is the predicted value of target variable, N is the total number of targets. The resulting MSE and MAE on training dataset are presented in the Table 1. Minimum MAE(ε_{xc}) and MSE(ε_{xc}) are obtained on benzene, maximums are achieved on silicon. MSE error on boundary conditions is on the order of 10^{-10}, which is considerably less than the total MSE error on the train dataset.

You can see in Fig. 3 that the distributions of silicon are significantly different from the overall distributions of the training dataset. One can see that crystalline silicon has relatively small electron density gradients and no region of low electron density due to its periodic structure. Therefore we attribute the maximum error of ε_{xc} in silicon to the fact that its characteristic values of electron density and its derivatives effectively have a smaller fraction in the training dataset compared to benzene and ammonia.

We also separately analyzed the spatial distribution of the error using benzene as an example. Figure 4 shows the distribution of the relative local error $|(\hat{x} - x)/x| \times 100\%$ in the benzene plane for both ε_{xc} and V_{xc}. The spatial distribution of the error clearly shows that it has the highest values in the region of the nuclei and at the boundary. Again, we associate the high local error in these places with a few examples of this type of data in its training set. If there were more extreme examples with high and low density in the training dataset, then the errors in the nuclei region and at the border should have dropped.

Testing of neural network.

We incorporate the developed NN XC functional into Octopus code and perform self-consistent cycle calculations. All calculations are converged, and relative total and XC energies errors are calculated. As a reference, we take PBE functional that was used for the training of NN. Results are presented in the Table 2. One can see that the highest error is achieved for atomic and molecular oxygen.

Substance	Error (E_{total}), %	$Error(E_{xc})$, %
C	0.052	0.287
S	0.015	0.102
SH	0.021	0.007
Cl	0.010	0.035
OH	0.554	1.036
Cl_2	0.001	0.035
O	1.556	4.029
P	0.008	0.071
O_2	0.646	2.482
PH	0.014	0.050
PH_2	0.011	0.023
S_2	0.023	0.040
Si	0.003	0.253

Table 2. Results of testing NN XC functional on a subset of IP13/03 dataset. E_{total} and E_{xc} denote total energy and exchange-correlation energy correspondingly obtained after convergence.

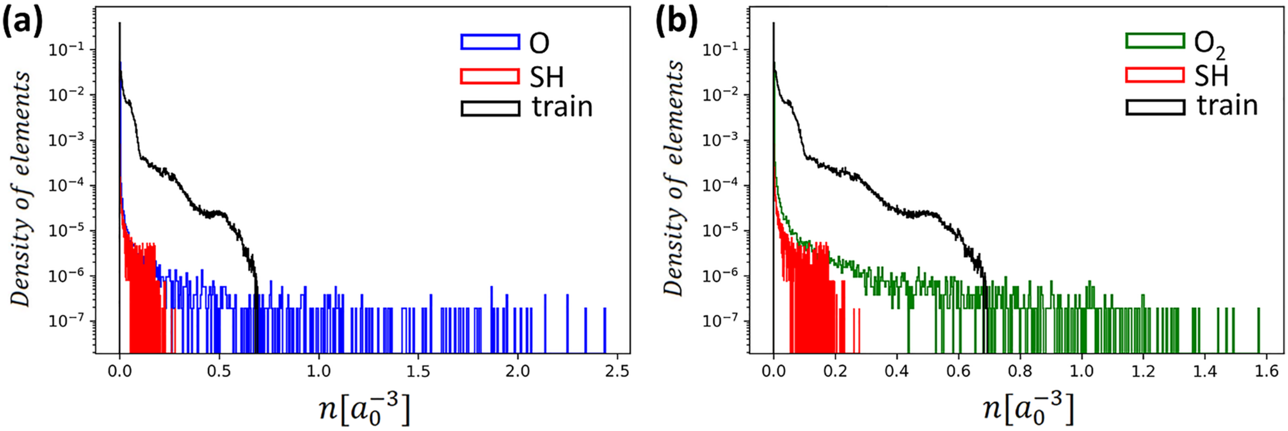

Figure 5. Comparison of distribution for non self-consistent electron density of atomic oxygen (**a**), molecular oxygen (**b**), hydrogen sulfide and self-consistent electron density from training (a_0 denotes the Bohr radius). One can see that atomic and molecular oxygen have a significant fraction of samples out of the train distribution. The step of the self-consistent cycle with maximum electron density was chosen for all substances. This example emphasizes the importance of including a wide range of electron density and derivatives in the training dataset because extreme values could occur during the self-consistent cycle.

The reasons for the error on oxygen were analyzed. It turned out that the main reason for such a discrepancy on some substances, including oxygen, is the intermediate densities that occurred in the process of a self-consistent cycle that the neural network did not see during the training process. The problem is that such densities can be arbitrarily large. The distributions of maximum electron densities in some cases are presented in Fig. 5 together with the training distribution. Analytical exchange-correlation functional contains asymptotics at infinity, which makes it possible to process such cases correctly. However, only boundary conditions at zero were used during our training cycle, but asymptotics at infinity were not used. This critical fact describes obtained discrepancy and gives hope for the further development and improvement of neural network exchange-correlation functionals by including such asymptotics[38,39].

To show the performance of developed NN XC on a chemically meaningful quantity we included results for Alkisomer11[11] dataset (see Table 3), for which we made PBE vs NN XC comparison of isomerization energies. Our model demonstrates results comparable to PBE, mean absolute error (MAE) for PBE is 1.6 kcal/mol and MAE for NN XC is 1.7 kcal/mol, which is a quite promising result. As it turned out as a result of the analysis, large errors on octane isomers even for PBE are explained by the fact that the corresponding geometries published in the article[11] are not completely relaxed.

To verify the importance of nonequilibrium densities for convergence of the SCF cycle we expanded the training dataset to include higher densities typical for intermediate densities of oxygen atom (O), which are obtained in self-consistent cycles (see Fig. 5). We stopped SCF iteration at 2, 4, 6, 8, 10, 12, 14, 16, 18, and 20 iterations and added 10 corresponding SCF results to the training set. We used spacing 0.273 a_0 and 64 × 64 × 64 mesh. The resulting neural network performed quite better for O_2, the relative error of total energy drops from 0.646 to 0.023% and the relative error of XC energy drops from 2.5 to 0.17%. Thus, we also have shown that the inclusion

Isomerization reaction	IE (PBE)	IE (NN)	Abs error, kcal/mol
Butane → isobutane	− 0.996	− 1.090	0.094
Pentane → isopentane	− 0.788	− 0.491	0.297
Pentane → neopentane	− 2.315	− 1.843	0.472
Hexane → isohexane	− 2.563	− 2.071	0.492
Hexane → neohexane	− 2.516	− 1.924	0.592
Hexane → diisopropyl	− 1.626	− 0.896	0.730
Hexane → 3methylpentane	− 2.062	− 1.409	0.653
Heptane → isoheptane	− 3.218	− 2.916	0.302
Heptane → neoheptane	− 2.807	− 2.746	0.061
Octane → hexamethylethane	5.224	6.326	1.102
Octane → isooctane	0.725	1.292	0.567

Table 3. Results of testing NN XC functional on Alkisomer11 dataset. IE denotes isomerization energy.

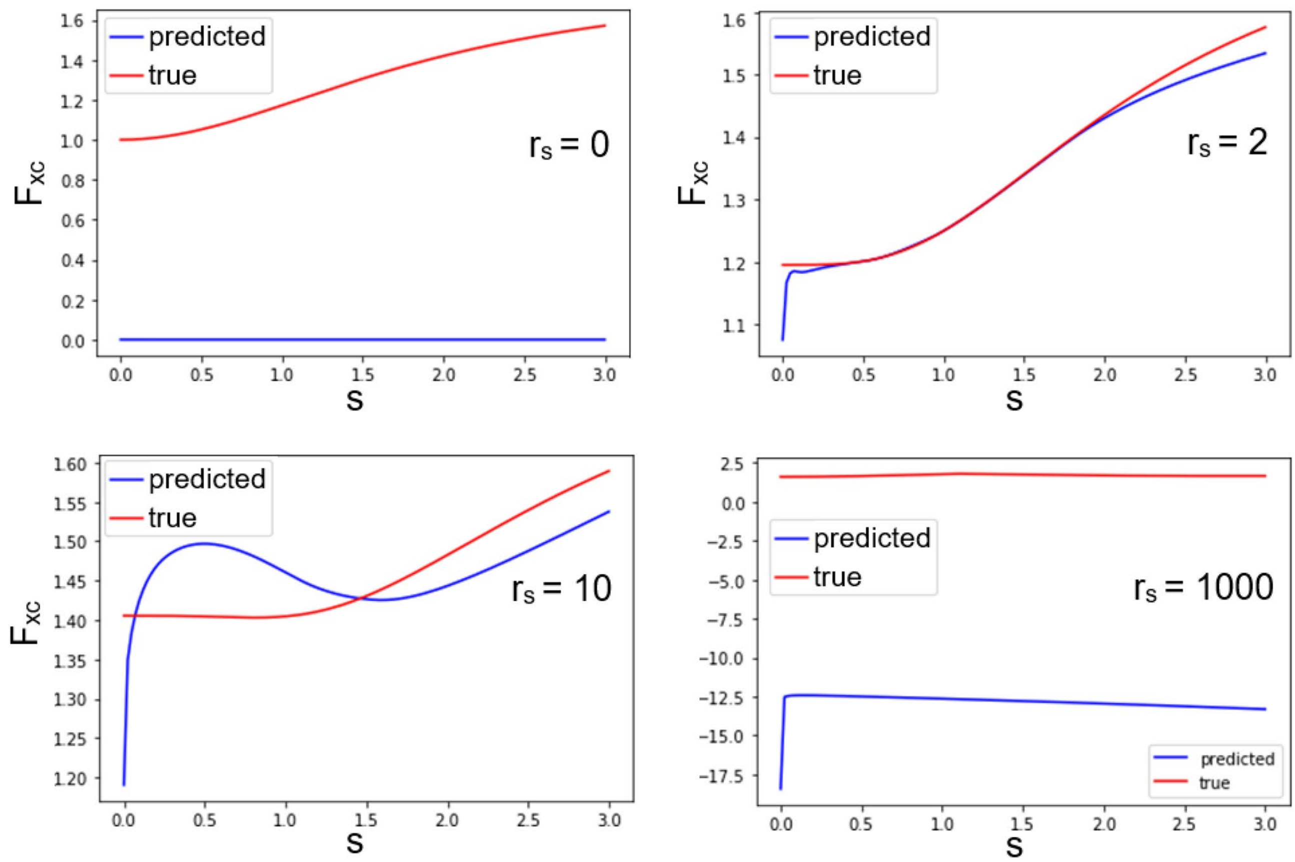

Figure 6. Enhancement factor F_{xc} at fixed values of r_s: 0, 2, 10, 1000. s is a reduced density gradient. There is a good agreement between the enhancement factors at $r_s = 2$ densities, which is close to the most frequent densities in the training set. Also one can see that NN XC functional is bad in guessing asymptotic in low ($r_s =$ 1000) and high ($r_s = 0$) limits of electron density.

of additional data, such as nonequilibrium densities (intermediate densities in the SCF cycle), makes it possible to increase the accuracy of the XC functional.

Enhancement factor $F_{xc}(r_s, s) = \varepsilon_{xc}/\varepsilon_x^{LDA}$ obtained from NN-E prediction was also analyzed, where s is a reduced density gradient $s = |\nabla \mathrm{n}|/2\mathrm{k_F n} = \frac{\left(\frac{1}{n}\right)^{4/3}|\nabla n|}{2\sqrt[3]{3}\pi^{2/3}}$. It compared with corresponding F_{xc} obtained from prediction of analytical exchange-correlation functional (PBE) by libxc (see Fig. 6). There is a high similarity between the enhancement factors at densities that closely match the most frequent densities in the training set ($r_s = 2$). We observe slightly less similarity at $r_s = 10$. This can be explained by the decreased number of elements with such a density in the training set. A significant discrepancy is observed in the limiting cases of high and low densities. In the case of high density (small r_s), this is explained by the absence of asymptotic at infinity. Despite

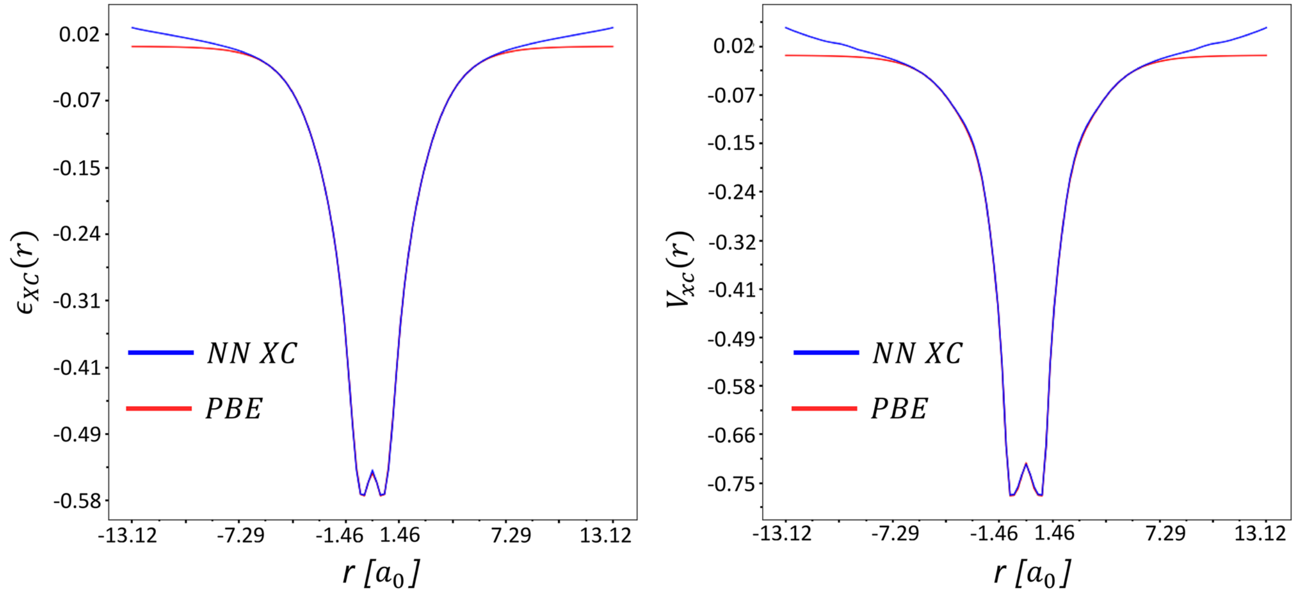

Figure 7. Slices of ε_{xc} and V_{xc} along the line connecting the atoms of a hydrogen molecule. Blue lines represent values predicted by our NN, red lines—values obtained by Octopus (where V_{xc} is correct functional derivative).

Rotation (deg.)	Total energy (PBE)	Total energy (NN)	XC energy (PBE)	XC energy (NN)
0	− 10.69546	− 10.69320	− 2.48893	− 2.48901
11.25	− 10.69546	− 10.69321	− 2.48891	− 2.48911
22.5	− 10.69542	− 10.69314	− 2.48890	− 2.48892
45	− 10.69544	− 10.69314	− 2.48892	− 2.48905
90	− 10.69546	− 10.69321	− 2.48893	− 2.48901

Table 4. Comparing NN and PBE energies of rotated SH molecule. All energies in Hartree.

the presence of asymptotic at zero densities in the process of training a neural network, the obtained result shows that this asymptotic may not be enough for obtaining a good enhancement factor.

Even though our model correctly reproduces the energies, the true exchange-correlation functional must satisfy many important physical constraints. Therefore, it is very crucial to discuss how such constraints can be added to the model and learning process in our future work. Firstly, it is important to mention that constraints use total exchange or correlation energies, so correct implementation of constraints can be effectively introduced only while training the model on total energies (much better, reaction energies). For example, an important property of such a functional is the density scaling property. In the article[40], it was shown that this property can be effectively taken into account using contrastive representation learning. Concerning our model, this means that the densities and all values obtained from it should be fed into NN-E and NN-V not in a “pure” form, but transformed using an encoder trained in an unsupervised way. As stated in the article mentioned above the NT-xent loss is a good choice. This loss causes the projected representations of unscaled and scaled densities of the same molecule to get close and the representations of different molecules to be far from each other. With another module predicting scaling factors this approach can help to incorporate uniform density scaling constraint. In the case of our work, we did not actively investigate the imposition of any constraints, this is the topic of ongoing research, but we checked that the presented functional does not lose accuracy when the SH molecule was rotated (see Table 4). As a result of additional calculations, it turned out that our model demonstrates excellent rotational equivariance with a total energy difference significantly smaller than 1 kcal/mol.

XC energy density and the XC potential are related with a functional derivative (see for example Eq. 1) Therefore, we compared the exchange-correlation potential predicted by the NN-V and the corresponding potential calculated using the functional derivative. Indirectly, the fact that the analytical relationship between ε_{xc} and V_{xc} is preserved can be shown by the fact that the errors both on energy and on potential are quite small at the same time (see Table 1). If the neural network did not correctly reproduce this connection, then, accordingly, the error on the potential would be quite large. We aslo demonstrated these results clearly on the hydrogen molecule and built distributions of ε_{xc} and V_{xc} along the line connecting the hydrogen atoms (see Fig. 7).

Conclusions

The second NN (such as NN-V) that directly predicts the XC potential is avoided in recent publications[26,41], by imposing automatic differentiation of the NN. Automatic differentiation is a remarkable property of neural networks, which is actively used for developing neural network exchange-correlation potentials. However, it has some disadvantages: if the neural network is deep enough then automatic differentiation makes much noise, especially if we talk about higher-order derivatives (most of the implementations in literature use DFT codes based on atomic orbitals, where only first-order derivatives are needed); for simple features, it is pretty straightforward to write an analytical form of the relationship between the potential and the XC energy. However, if we take complex and non-standard features (semi-local,global ones), it can be practically impossible to connect the potential and the exchange-correlation energy. This is where the advantage of the second neural network becomes important—it provides an alternative path for an approximation that does not involve taking partial derivatives. This allows using more complex features in the machine learning pipeline.

Low error on the validation dataset indicates that the developed approach to the architecture of the XC functional interpolates well the existing XC functionals and has the high generalizing ability. Furthermore, in the framework of the proposed architecture, one can train NN-E and NN-V separately, making it flexible to use. The basic strategy to create working functional in the framework of the proposed architecture would be initially to train the NN-V part on a specific type of existing XC functional such as LDA, GGA, or meta-GGA, using various types of input features. Then the NN-E part of the neural network is trained on the data obtained by accurate post-Hartree–Fock or quantum Monte Carlo methods.

In this work, we did not take into account that the resulting functional may violate the rules of critical physical conditions[41,42]. However, all these conditions can be introduced by modifying the training loop or loss function and, therefore, can be explicitly taken into account. This is a matter for further research.

The main advantage of the NN approach in comparison with other interpolation techniques for XC functionals is its flexibility to incorporate exchange-correlation data from different sources, such as post-Hartree–Fock and quantum Monte Carlo. It is possible that application of the NN to interpolate high-level XC quantum data could eliminate many heuristics used in the traditional construction of XC functionals.

Data availability

Optimized NN weights and code to use it in calculations are available at https://github.com/AlexanderFreeman/octopus-nn-2.

Received: 21 May 2022; Accepted: 4 August 2022
Published online: 19 August 2022

References

1. Hohenberg, P. & Kohn, W. Inhomogeneous electron gas. *Phys. Rev.* **136**, B864 (1964).
2. Kohn, W. & Sham, L. J. Self-consistent equations including exchange and correlation effects. *Phys. Rev.* **140**, A1133 (1965).
3. Ceperley, D. M. & Alder, B. Ground state of the electron gas by a stochastic method. *Phys. Rev. Lett.* **45**, 566 (1980).
4. Zhao, Y., Schultz, N. E. & Truhlar, D. G. Exchange-correlation functional with broad accuracy for metallic and nonmetallic compounds, kinetics, and noncovalent interactions (2005).
5. Vosko, S. H., Wilk, L. & Nusair, M. Accurate spin-dependent electron liquid correlation energies for local spin density calculations: A critical analysis. *Can. J. Phys.* **58**, 1200–1211 (1980).
6. Perdew, J. P. & Zunger, A. Self-interaction correction to density-functional approximations for many-electron systems. *Phys. Rev. B* **23**, 5048 (1981).
7. Perdew, J. P. & Wang, Y. Accurate and simple analytic representation of the electron-gas correlation energy. *Phys. Rev. B* **45**, 13244 (1992).
8. Wang, Y. & Perdew, J. P. Spin scaling of the electron-gas correlation energy in the high-density limit. *Phys. Rev. B* **43**, 8911 (1991).
9. Perdew, J. P. *et al.* Atoms, molecules, solids, and surfaces: Applications of the generalized gradient approximation for exchange and correlation. *Phys. Rev. B* **46**, 6671 (1992).
10. Perdew, J. P., Burke, K. & Ernzerhof, M. Generalized gradient approximation made simple. *Phys. Rev. Lett.* **77**, 3865 (1996).
11. Mardirossian, N. & Head-Gordon, M. Thirty years of density functional theory in computational chemistry: An overview and extensive assessment of 200 density functionals. *Mol. Phys.* **115**, 2315–2372 (2017).
12. Lani, G., Di Marino, S., Gerolin, A., van Leeuwen, R. & Gori-Giorgi, P. The adiabatic strictly-correlated-electrons functional: Kernel and exact properties. *Phys. Chem. Chem. Phys.* **18**, 21092–21101 (2016).
13. Maier, T. M., Haasler, M., Arbuznikov, A. V. & Kaupp, M. New approaches for the calibration of exchange-energy densities in local hybrid functionals. *Phys. Chem. Chem. Phys.* **18**, 21133–21144 (2016).
14. Mori-Sánchez, P. & Cohen, A. J. The derivative discontinuity of the exchange-correlation functional. *Phys. Chem. Chem. Phys.* **16**, 14378–14387 (2014).
15. Mori-Sánchez, P. & Cohen, A. J. Exact density functional obtained via the levy constrained search. *J. Phys. Chem. Lett.* **9**, 4910–4914 (2018).
16. Needs, R., Towler, M., Drummond, N. & Ríos, P. L. Continuum variational and diffusion quantum Monte Carlo calculations. *J. Phys. Condens. Matter* **22**, 023201 (2009).
17. Kolorenč, J. & Mitas, L. Applications of quantum Monte Carlo methods in condensed systems. *Rep. Prog. Phys.* **74**, 026502 (2011).
18. Cremer, D. Møller–Plesset perturbation theory: From small molecule methods to methods for thousands of atoms. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **1**, 509–530 (2011).
19. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control Signals Syst.* **2**, 303–314 (1989).
20. Tozer, D. J., Ingamells, V. E. & Handy, N. C. Exchange-correlation potentials. *J. Chem. Phys.* **105**, 9200–9213 (1996).
21. Nagai, R., Akashi, R., Sasaki, S. & Tsuneyuki, S. Neural-network Kohn–Sham exchange-correlation potential and its out-of-training transferability. *J. Chem. Phys.* **148**, 241737 (2018).
22. Nagai, R., Akashi, R. & Sugino, O. Completing density functional theory by machine learning hidden messages from molecules. *NPJ Comput. Mater.* **6**, 1–8 (2020).
23. Lei, X. & Medford, A. J. Design and analysis of machine learning exchange-correlation functionals via rotationally invariant convolutional descriptors. *Phys. Rev. Mater.* **3**, 063801 (2019).
24. Ramos, P. & Pavanello, M. Static correlation density functional theory. arXiv preprint arXiv:1906.06661 (2019).

25. Ryabov, A., Akhatov, I. & Zhilyaev, P. Neural network interpolation of exchange-correlation functional. *Sci. Rep.* **10**, 1–7 (2020).
26. Li, L. *et al.* Kohn-sham equations as regularizer: Building prior knowledge into machine-learned physics. *Phys. Rev. Lett.* **126**, 036401 (2021).
27. Kirkpatrick, J. *et al.* Pushing the frontiers of density functionals by solving the fractional electron problem. *Science* **374**, 1385–1389 (2021).
28. Perdew, J. P., Burke, K. & Ernzerhof, M. Generalized gradient approximation made simple [phys. rev. lett. 77, 3865 (1996)]. *Phys. Rev. Lett.* **78**, 1396–1396 (1997).
29. Andrade, X. *et al.* Real-space grids and the octopus code as tools for the development of new simulation approaches for electronic systems. *Phys. Chem. Chem. Phys.* **17**, 31371–31396 (2015).
30. Andrade, X. *et al.* Time-dependent density-functional theory in massively parallel computer architectures: The octopus project. *J. Phys. Condens. Matter* **24**, 233202 (2012).
31. Andrade, X. & Aspuru-Guzik, A. Real-space density functional theory on graphical processing units: Computational approach and comparison to Gaussian basis set methods. *J. Chem. Theory Comput.* **9**, 4360–4373 (2013).
32. Lynch, B. J. & Truhlar, D. G. Robust and affordable multicoefficient methods for thermochemistry and thermochemical kinetics: the mccm/3 suite and sac/3. *J. Phys. Chem. A* **107**, 3898–3906 (2003).
33. Balbás, L., Martins, J. L. & Soler, J. M. Evaluation of exchange-correlation energy, potential, and stress. *Phys. Rev. B* **64**, 165110 (2001).
34. Lehtola, S., Steigemann, C., Oliveira, M. J. & Marques, M. A. Recent developments in libxc-a comprehensive library of functionals for density functional theory. *SoftwareX* **7**, 1–5 (2018).
35. Schlipf, M. & Gygi, F. Optimization algorithm for the generation of oncv pseudopotentials. *Comput. Phys. Commun.* **196**, 36–44 (2015).
36. Paszke, A. *et al.* Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32* (eds. Wallach, H. *et al.*) 8024–8035 (Curran Associates, Inc., 2019).
37. Clevert, D.-A., Unterthiner, T. & Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus) (2016).
38. Cuierrier, E., Roy, P.-O. & Ernzerhof, M. Constructing and representing exchange-correlation holes through artificial neural networks. *J. Chem. Phys.* **155**, 174121 (2021).
39. Nagai, R., Akashi, R. & Sugino, O. Machine-learning-based exchange-correlation functional with physical asymptotic constraints. arXiv preprint arXiv:2111.15593 (2021).
40. Gong, W. *et al.* Incorporation of density scaling constraint in density functional design via contrastive representation learning. arXiv preprint arXiv:2205.15071 (2022).
41. Schmidt, J., Benavides-Riveros, C. L. & Marques, M. A. Machine learning the physical nonlocal exchange-correlation functional of density-functional theory. *J. Phys. Chem. Lett.* **10**, 6425–6431 (2019).
42. Gaiduk, A. P. & Staroverov, V. N. How to tell when a model Kohn–Sham potential is not a functional derivative. *J. Chem. Phys.* **131**, 044107 (2009).

Acknowledgements

The research was partly carried out using supercomputers at the Joint Supercomputer Center of the Russian Academy of Sciences (JSCC RAS) and the Skoltech CDISE supercomputer "Zhores".

Author contributions

A.R. and P.Z. performed research. A.R. and P.Z. wrote the paper. P.Z. and I.A. supervised the project. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to A.R.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.