Machine learning the derivative discontinuity of density-functional theory

Johannes Gedeon (johannes-gedeon@web.de)
Institut für Physik, Martin-Luther-Universität Halle-Wittenberg
Jonathan Schmidt
Institut für Physik, Martin-Luther-Universität Halle-Wittenberg
Matthew Hodgson
Department of Physics, Durham University https://orcid.org/0000-0002-2256-6860
Jack Wetherel
Ecole Polytechnique, CNRS, Institut Polytechnique de Paris
Carlos Benavides-Riveros
Martin-Luther-Universität Halle-Wittenberg https://orcid.org/0000-0001-6924-727X
Miguel Marques
Martin-Luther-Universität Halle-Wittenberg https://orcid.org/0000-0003-0170-8222

Article

Keywords: machine learning, density functional theory, formulation

Posted Date: July 13th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-677067/v1

Version of Record: A version of this preprint was published at Machine Learning: Science and Technology on October 20th, 2021. See the published version at https://doi.org/10.1088/2632-2153/ac3149.

Machine learning the derivative discontinuity of density-functional theory

(Dated: July 5, 2021)

Machine learning is a powerful tool to design accurate, highly non-local, exchange-correlation functionals for density functional theory. So far, most of those machine learned functionals are trained for systems with an integer number of particles. As such, they are unable to reproduce some crucial and fundamental aspects, such as the explicit dependency of the functionals on the particle number or the infamous derivative discontinuity at integer particle numbers. Here we propose a solution to these problems by training a neural network as the universal functional of density-functional theory that (i) depends explicitly on the number of particles with a piece-wise linearity between the integer numbers and (ii) reproduces the derivative discontinuity of the exchange-correlation energy. This is achieved by using an ensemble formalism, a training set containing fractional densities, and an explicitly discontinuous formulation.

INTRODUCTION

In their now famous paper, Hohenberg and Kohn proved that the electron density $\rho(\boldsymbol{r})$ suffices to compute all observables of a system of interacting electrons [1]. Due to a remarkable balance of computational cost and numerical precision, first principles modeling of electronic systems based on this density functional theory (DFT) is nowadays a daily practice, with great impact in material science, quantum chemistry or condensed matter [2]. The success of DFT is to a large extent based on the Kohn-Sham formulation, that utilizes a system of non-interacting electrons that has the same density as the interacting one [3]. The main ingredient of this formulation is $E_{\rm xc}[\rho]$, the universal exchange-correlation (xc) functional, whose functional derivative provides an effective external potential for the non-interacting particles. Yet, while the Hohenberg-Kohn theorem proves the uniqueness of such a functional, it does not give any indication regarding its specific form. To circumvent this issue, a very large number of approximate functionals were developed in the last decades [4, 5], often combining empirical knowledge, exact mathematical conditions, and a great deal of ingenuity.

Inspired by the success of machine learning (ML) in various technological applications, including image and speech recognition [6], the last couple of years have seen the development of several neural-network-based approximations to $E_{\rm xc}[\rho]$. Indeed, it is now firmly established that machine-learning offers a new generation of accurate, highly non-local, xc functionals [7]. While those functionals are designed to perform tasks of different degree of complexity, all share the aim of learning one of the maps of DFT, namely, the Hohenberg-Kohn map between the external potential $v(\boldsymbol{r})$ and the density $\rho(\boldsymbol{r})$ [8–12], or the Kohn-Sham map between the density $\rho(\boldsymbol{r})$ and the xc functional $E_{\rm xc}[\rho(\boldsymbol{r})]$ and its functional derivative $v_{\rm xc}[\rho(\boldsymbol{r})] = \delta E_{\rm xc}[\rho(\boldsymbol{r})]/\delta\rho(\boldsymbol{r})$ [13–15].

The functionals delivered by machine-learning DFT (ML-DFT) are in general non-local, in the sense that they use multiple density points as input, and can be efficiently trained with data from reference methods. Yet, since ML-DFT functionals are mostly trained in Hilbert spaces with an *integer* number of particles, they are still unable to reproduce some critical and fundamental aspects of DFT. For instance, it is known that any satisfactory definition of the energy functional must depend explicitly on the particle number [16–18]. Furthermore, the derivative of the xc functional in terms of the number of particles exhibits a discontinuity that plays a crucial role in the description of electronic bandgaps [19–22], charge-transfer excitations [23, 24], molecular dissociation [25–28], or even Mott insulators [29], to name but a few examples.

Systems with non-integer (fractional) number of electrons ($N+\epsilon$) are defined as statistical mixtures of systems with integer number of particles [20, 30]. As such, the density $\rho_{N+\epsilon}(\boldsymbol{r})$ and total energy $E(N+\epsilon)$ are piecewise linear functions of ϵ, namely:

$$\rho_{N+\epsilon}(\boldsymbol{r}) = (1-\epsilon)\rho_N(\boldsymbol{r}) + \epsilon\rho_{N+1}(\boldsymbol{r}), \tag{1a}$$

$$E(N+\epsilon) = (1-\epsilon)E(N) + \epsilon E(N+1) \tag{1b}$$

with $0 \leq \epsilon \leq 1$. At integer N (i.e., when $\epsilon = 0$), the derivatives of the density and the energy exhibit a discontinuity, and the xc potential $v_{\rm xc}(\boldsymbol{r})$ jumps by a finite value [31]. The difference in the slope on the left/right side of the total energy at integer values is equal to the fundamental gap [20]:

$$I - A = \left.\frac{\partial E}{\partial N}\right|_+ - \left.\frac{\partial E}{\partial N}\right|_- , \tag{2}$$

where I is the ionization energy and A the electron affinity. Yet, in practice, standard approximations to the xc functionals that depend explicitly on the electronic density, such as the local-density (LDA) and generalized-gradient (GGA) approximations, are continuously differentiable functions of N and lack therefore a derivative discontinuity. Meta-GGAs can exhibit a discontinuity due to their dependence on the kinetic-energy density, but it is usually too small or even negative [32]. Due to their dependence on the Kohn-Sham orbitals, orbital functionals are discontinuous [33], but this comes at the price of a much higher computational effort.

In addition to the discontinuity, a universally useful approximation for the xc functional must be "N-electron

self-interaction-free" for *all* positive integer N [34], meaning that the total energy of a system with $N+\epsilon$ electrons in the range $(N,\, N+1)$ should exhibit a linear variation with respect to ϵ. For attractive interactions the energy is a convex function with straight lines joining subsets of ground-state energies [35]. Yet, approximate functionals deviate from such a correct behavior. It has been shown that semi-local density functionals are in general convex with perhaps small concave pieces [36]. Even the Hartree Fock theory leads to piecewise concave curves between integers [36]. We note that the relatively well-defined curvature of the curves is ultimately the reason for the success of the Slater half-occupation scheme [37] or the LDA-1/2 method [38]. In fact, these schemes use the derivative at the midpoint (i.e., at $N-0.5$), that can be shown to be equal to the slope of the straight line between $E(N-1)$ and $E(N)$ if the curvature is constant, irrespective of its sign [39]. The centrality of these pressing issues in DFT can be further highlighted by the fact that a rigorous description of the delocalization error can be related with the energy curve of the xc functionals lying below the straight energy lines [36, 40].

In this work, we propose a way to train a neural network as the *ensemble* universal functional of a system of fractional electron numbers that describes correctly the derivative discontinuity and the piecewise linear behavior. The ML functionals we present contain explicitly the physics of the derivative discontinuity of DFT, are highly non-local, and are trained for systems with fractional densities. For this reason, our functionals can potentially address the well known delocalization and static correlation errors of DFT [41–43] simultaneously.

RESULTS AND DISCUSSION

Inspired by the neural network topology proposed in Ref. [13], our neural network takes an electronic density as an input and returns the corresponding xc energy, whose functional derivative can in turn be used to solve the Kohn-Sham equations. The network is a sliding window convolution (SWC) network. For a 1D system of discrete spatial points $\{r_1, ..., r_W\}$, a *window* with a certain *kernel size* κ scans each data point $\rho_\sigma(r_j)$ and its $\kappa-1$ nearest neighbors $\eta_\sigma(r_j,\kappa) = \{\rho_\sigma(r_{j-(\kappa-1)/2}), ..., \rho_\sigma(r_{j+(\kappa-1)/2})\}$ with $\sigma = \uparrow, \downarrow$ to calculate a local energy $\epsilon^\theta_{\rm loc}\big[\eta_\uparrow(r_j,\kappa), \eta_\downarrow(r_j,\kappa)\big]$. The total xc energy is calculated by summing over the local energies:

$$E_{\rm xc}\left[\rho_\uparrow, \rho_\downarrow\right] = \sum_j \rho(r_j)\, \epsilon^\theta_{\rm loc}\big[\eta_\uparrow(r_j,\kappa), \eta_\downarrow(r_j,\kappa)\big]\,. \qquad (3)$$

Here, θ denotes the trainable parameters. The input channels can be the total electronic density $\rho = \rho_\uparrow + \rho_\downarrow$ or the spin densities $\rho_\uparrow$ and $\rho_\downarrow$. The corresponding xc potentials can be computed using automatic differentiation, as shown in Ref. [13]. The parameters of the neural network are updated according to the loss function:

$$\mathcal{L}(\theta;\alpha,\beta) = \alpha\,{\rm MSE}(v^\uparrow_{\rm xc}[\rho_\uparrow,\rho_\downarrow], v^\downarrow_{\rm xc}[\rho_\uparrow,\rho_\downarrow]) + \beta\,{\rm MSE}(E_{\rm xc}[\rho_\uparrow,\rho_\downarrow])\,, \qquad (4)$$

where α and β are *fixed* weights that can be adjusted to expedite convergence, and MSE is the mean squared error. A more detailed description of our networks can be found in the Methods section.

We improve the performance of this architecture by (i) training our neural network with non-integer densities, (ii) introducing the jump of the xc potential at integer numbers into the loss function, and (iii) adding an explicit discontinuity at integer electron numbers. In the following we discuss the details of these new approaches.

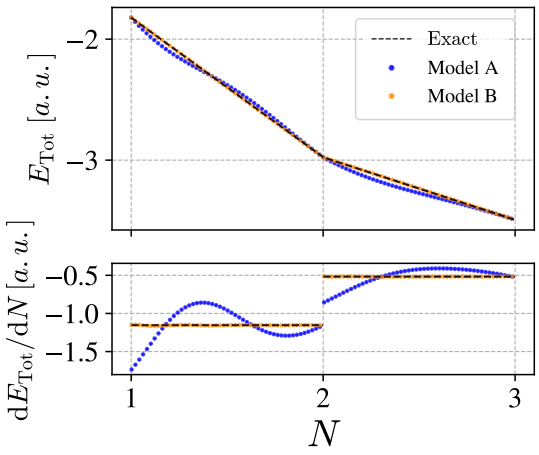

FIG. 1. Comparison of the total energy for a certain external potential (top) as well as its derivative with respect to the particle number (bottom) for two different models: Model A has been trained with integer densities only. Model B has been trained with fractional densities, the exact $\Delta_{\rm xc}$-shift, and employs the AF (see text).

Fractional particle numbers

In general, neural networks do not extrapolate well outside the distribution of the samples used for their training . Consequently, we can not expect that machines trained solely for integer densities (as usually done) will exhibit the correct linear behaviour of the energy. To illustrate this behavior we plot, in Fig. 1, the total energy calculated with a neural network trained solely with integer densities (model A in the figure) as a function of the number of particles for a 1-dimensional system. Besides the fact that model A is far from linear, we can also note that the sign of the curvature is not constant, with both concave and convex parts. This is somehow to be

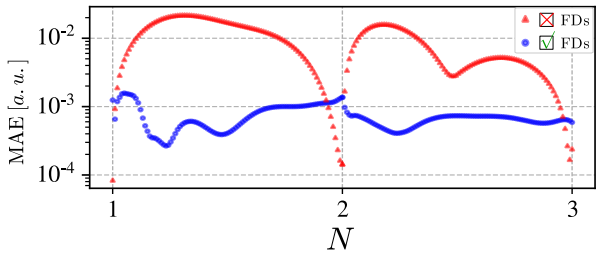

FIG. 2. Comparison of the mean absolute error (MAE) of the total energy between two different models (averaged over 100 external potentials). The model "FDs" (blue) includes fractional densities $\epsilon = (0.05, 0.2, 0.5, 0.8, 0.95)$ in the training. The second model (red) includes only integer densities in the training. Both models were trained within the spin-DFT framework.

expected, as the network, in contrast to the usual xc functionals, only incorporates physical knowledge through the training examples with integer densities. As such, we can easily see that approaches such as LDA-1/2 are bound to fail in this case.

As a first strategy to solve this problem, we decided to include samples calculated within ensemble-DFT at fractional densities in our training. To obtain this data we created a set of total energies and electronic densities for a series of 1-dimensional exact calculations. We then constructed ensemble densities and inverted the ensemble Kohn-Sham equations, in order to compute the exact xc energy and potential for these systems. We used an inversion algorithm based on Ref. [44] that we extended for both spin-DFT [45] and to ensemble systems. As a result, we created exact training and testing data with particle numbers between 1 and 3 electrons, that we used to train our models.

In Fig. 2 we compare the mean absolute error (MAE) of the total energy for a functional trained with fractional densities (blue) and a functional trained only for systems with integer densities (red) (averaged over a test set of 100 external potentials). As we can see, the model trained at integer densities yields an excellent prediction for the total energy at those integers, but exhibits a considerably larger error at fractional numbers. Remarkably, by simply adding fractional densities to the training set, the error decreases by more than one order of magnitude, and the MAE over the entire $[1, 3]$-range remains below 2×10^{-3} a.u. We note in passing that we trained the networks both for spin-restricted and spin-unrestricted DFT with similar results. As such, we decided to use the spin-restricted formalism in the following.

While this strategy resolved, to a large extent, the many-electron self-interaction error, a problem still remains in the vicinity of the integer particle numbers. In fact, our network is fully differentiable, and does not (in fact can not) exhibit a true derivative discontinuity (in a mathematical sense) as a function of N.

Jump in the xc potential

We can easily relate the discontinuity of the total energy at integer particle numbers with an uniform shift in the potential. Indeed, the exact uniform shift Δ_{xc}^N of v_{xc} at integer particle number N obeys the relation [46]:

$$\left.\frac{\partial E}{\partial N}\right|_+ - \left.\frac{\partial E}{\partial N}\right|_- = \varepsilon_{\mathrm{s}}^N + \Delta_{\mathrm{xc}}^N \,, \tag{5}$$

where $\varepsilon_{\mathrm{s}}^N$ is the Kohn-Sham gap, i.e. the difference between the lowest unoccupied (LUMO) and the highest occupied (HOMO) molecular orbital energies. Noticeably, $E(N)$ and $E(N \pm 1)$, as well as the eigenvalues corresponding to the LUMO and HOMO can be computed while creating the training sets.

Our second strategy consists of computing, in our learning process, both $\rho_{N+\epsilon}$, and the exact shift $v_{\mathrm{xc}}(N_+) - v_{\mathrm{xc}}(N_-)$. The corresponding mean squared error is then used to extend the loss function in Eq. (4)

$$\mathcal{L} \to \mathcal{L} + \lambda \,\mathrm{MSE}(v_{\mathrm{xc}}(N+\epsilon) - v_{\mathrm{xc}}(N)) \,, \tag{6}$$

where λ is an additional hyperparameter.

We expect that this extended loss function can help the functional to learn the correct shift in the derivative. Unfortunately training our basic SWC network with this loss function failed for any learning rate tested. This can be understood from the fact that our network is still fully differentiable, and can not be forced to learn a discontinuous function. It is clear that to resolve this issue we have to allow explicitly for a discontinuous behaviour in the neural network topology.

Incorporating the discontinuity

An intuitive way to introduce a discontinuity in the derivatives of the neural network is to use non-differentiable activation functions (e.g., the rectified linear unit [47]). But there is no obvious reason why a non-differentiability at integer particle numbers will appear – and these networks will most likely become non-differentiable with respect to the density ρ. To overcome this problem we take an alternative route: We define an *auxiliary function* (AF), which we force to be non-differentiable at all integer particle numbers:

$$u\left[\rho_\uparrow, \rho_\downarrow\right] = a + \frac{|\sin(\pi\tilde{n})|}{\tilde{n} + b} \sum_j u_{\mathrm{loc}}^{\theta'}\left[\eta_\uparrow(r_j, \kappa), \eta_\downarrow(r_j, \kappa)\right] \,,$$

where $\tilde{n} = \int (\rho_\uparrow + \rho_\downarrow)\, \mathrm{d}r$ is the (fractional) number of particles and a and b are arbitrary positive constants. Notice that the non-differentiability of the AF comes from the non-differentiability of the function $|\sin(\pi\tilde{n})|$. The local functions $u_{\mathrm{loc}}^{\theta'}$ are obtained by using another SWC neural network. We then replace the functional $E_{\mathrm{xc}}\left[\rho_\uparrow, \rho_\downarrow\right]$

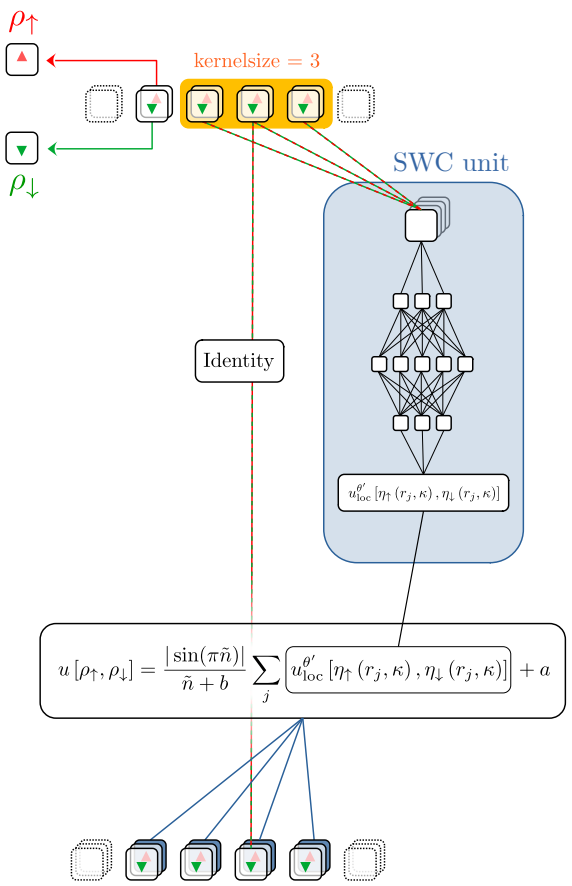

FIG. 3. Integration of the non-differentiable auxiliary function $u\left[\rho_\uparrow,\rho_\downarrow\right]$ as a third channel to the basic SWC unit. The input data contains two channels for both spin-densities $\rho_\uparrow$ and $\rho_\downarrow$ respectively.

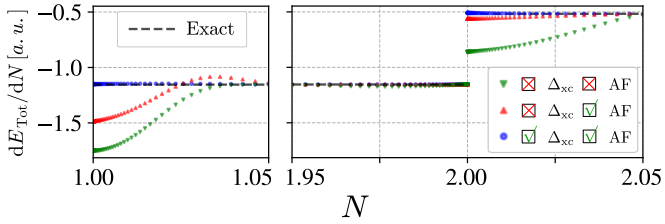

FIG. 4. Comparison of the derivatives of the energy with respect to the particle number N for three different models evaluated at a randomly chosen external potential: the first model uses only fractional densities, the second incorporates the exact xc shift in the loss function, and the third one uses a non-differentiable AF, in addition to the exact xc shift and the fractional densities.

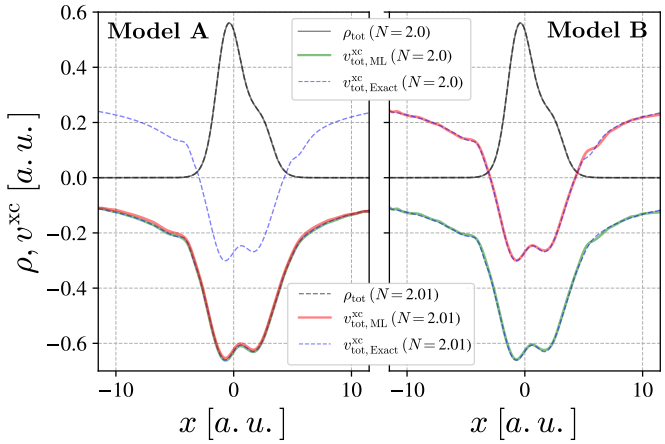

FIG. 5. Comparison between the predicted jump of the xc potential at $N=2$ for Model A and Model B (as explained in the main text). The results correspond to the same external potential used in Fig. 1.

in Eq. (3) by $E'_{\mathrm{xc}}[\rho_\uparrow,\rho_\downarrow]\equiv E_{\mathrm{xc}}\left[\rho_\uparrow,\rho_\downarrow,u\left[\rho_\uparrow,\rho_\downarrow\right]\right]$, where the additional channel has been appended to the spin channels. Here each point carries the (same) information of the value of the non-differentiable AF as illustrated in Fig. 3. As we show below, this procedure ensures that the machine can learn the non-differentiable function in an efficient manner.

To study the performance of our approach, Fig. 4 presents the predicted derivative of the total energy for three different training models: (1) a model trained only with fractional densities, (2) a model using the AF in the network architecture and trained with fractional densities, and (3) a model using the AF, the shift in the loss function and fractional densities. First, we should note that the inclusion of the AF solved the training problems discussed in the previous section. As expected, the first model does not predict the correct derivative discontinuity, despite yielding very good errors at fractional particle numbers. While the second model already demonstrates major improvements, especially at $N=2$, our most important finding is that the third model does exhibit a remarkable agreement with the exact results.

In Fig. 1 the energy as a function of the particle number is displayed for two models: (A) a model trained with integer densities only and (B) a model using fractional densities, the shift in the loss function and the AF. The situation is clearly much better for model B, as the energy is very close to linear between integers and shows a cusp at $N=2$. As shown in Fig. 5, this model is also able to reproduce the correct jump in the xc potential: Here we display the xc potentials and densities for a randomly selected external potential at 2 and 2.01 electrons. On the left panel of Fig. 5 we can see that the xc potential of the basic model A is correct at 2 electrons but barely changes when going from 2 to 2.01. Model B, on the other hand, shows the correct uniform shift in its xc potential.

CONCLUSIONS

We trained a neural network as an exchange correlation functional that (i) depends explicitly on the number of particles; (ii) yields total energies that are piecewise linear between the integers and (iii) reproduces the infamous derivative discontinuity of the exchange-correlation energy with remarkable accuracy. To do so we extended the sliding window convolution algorithm to systems with fractional number of particles, and developed a non-differentiable auxiliary function that allows the network to learn correctly the derivative discontinuity. The most efficient way to train a model yielding highly accurate predictions for the energy in the entire range of particle numbers is by (i) adding multiple fractional densities into the training data, (ii) training for the correct shift in the xc potential at integer particle numbers, and (iii) incorporating an auxiliary function with a discontinuous derivative with respect to the particle number at integers.

Our work pushes forward the on-going research on machine-learning functionals by incorporating the correct physics into the training process, thereby improving the path towards an exact functional [48]. As an outlook, we think it will be important to incorporate and test our results for realistic 3D systems. We also expect that our results can stimulate research on similar problems for ensemble DFT (understood as mixtures of ground and excites states), orbital free DFT or functional theories of reduced density matrices [49–53].

METHODS

In this section we present all methods and computational details necessary to arrive at our results. First we discuss the generation of the training data and second the details of the network implementation.

Exact calculations

To train and test our models we created a set of 1-dimensional exact calculations and Kohn-Sham inversions. We sampled 1500 external (Coulomb-) potentials and computed the exact electronic ground state densities as well as the corresponding energies by solving the eigenvalue problem for the electronic Hamiltonian

$$H = -\sum_{i=1}^{N}\frac{\nabla_i^2}{2} - \sum_{j=1}^{K}\sum_{i=1}^{N}\frac{Z_j}{\sqrt{1+|R_j-r_i|^2}} + \sum_{i<j}\frac{1}{\sqrt{1+|r_i-r_j|^2}}\,, \quad (7)$$

where K is the total number of nuclei, the variables R_j and Z_j denote the position and charge of the jth nuclei respectively, $N \in \{1,2,3\}$ is the total number of electrons, and r_i is the position of the ith electron. We solve the exact ground-state problem with Octopus [54], using a grid spacing of 0.1 a.u. and a box size of 23 a.u. (leading to a grid with 231 points). To circumvent the integrability problem of the Coulomb interaction in 1D we used a softened interaction. The total number of nuclei K were set to be 1, 2 or 3, such that their individual charges satisfy $\sum_k Z_k = 3$. Their positions were randomly distributed with $|R_k| \leq 4$ a.u.

Spin densities

As Octopus [54] only provides directly spin-densities and wave-functions for two-particle systems we now discuss the problem of obtaining the spin densities for the three particle systems.

We start by noticing that solving the eigenvalue problem $\hat{H}\Phi = E\Phi$ for the Hamiltonian (7) yields all many-particle solutions, including both fermionic and bosonic states. A spin-adapted fermionic solution of the form $\Psi(r_1\sigma_1,\ldots,r_N\sigma_N)$ can be obtained by projecting any spatial solution Φ on the Young diagrams belonging to certain spin quantum numbers (S, M), with S being the total spin and $M = \sum_i \sigma_i$ [55, 56]. For a detailed description we refer to [55, 57, 58]. Let us denote a set of f primitive, degenerated, but orthogonal spin functions as $\{X(N,S,M;i)\}_{i=1\cdots f}$. A permutation $\mathbf{P}$ acting on a spin function can be expressed as a linear combination of all primitive spin functions:

$$\mathbf{P}X(N,S,M;i) = \sum_{j=1}^{f} X(N,S,M;j)U(P)^S_{ji}\,. \quad (8)$$

The expansion coefficients $U(P)^S_{ji}$ can be calculated using the orthogonality of $X(N,S,M;j)$ [59]. By taking into account the antisymmetrization $\mathcal{A} = \frac{1}{\sqrt{N!}}\sum_P(-1)^P\mathbf{P}$ of the product of the spin and spatial parts $\Phi X(N,S,M;i)$ one obtains a sum of products of spatial and spin functions, as follows:

$$\begin{aligned}\Psi_i &= \mathcal{A}\Phi X(N,S,M;i)\\ &= \frac{1}{\sqrt{N!}}\sum_P(-1)^p\mathbf{P}^r\Phi\mathbf{P}^\sigma X(N,S,M;i)\\ &= \frac{1}{\sqrt{f}}\sum_{j=1}^{f}X(N,S,M;j)\Phi^S_{ji}\,, \end{aligned} \quad (9)$$

where $\mathbf{P}^r$ and $\mathbf{P}^\sigma$ denote that the permutations operates on spatial and spin coordinates respectively, and

$$\Phi^S_{ji} = \sqrt{\frac{f}{N!}}\sum_P U(P)^S_{ji}(-1)^p\mathbf{P}^r\Phi(\mathbf{r}_1,\ldots,\mathbf{r}_N)\,. \quad (10)$$

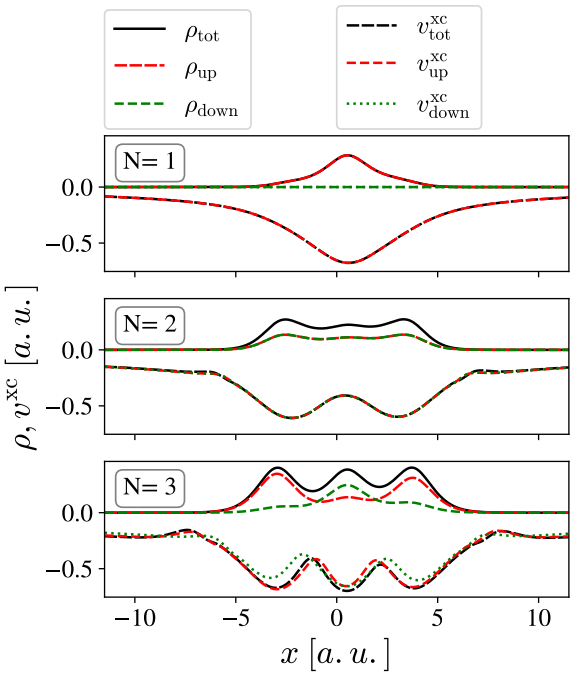

FIG. 6. Densities and corresponding xc potentials for a certain external potential for different integer particle numbers.

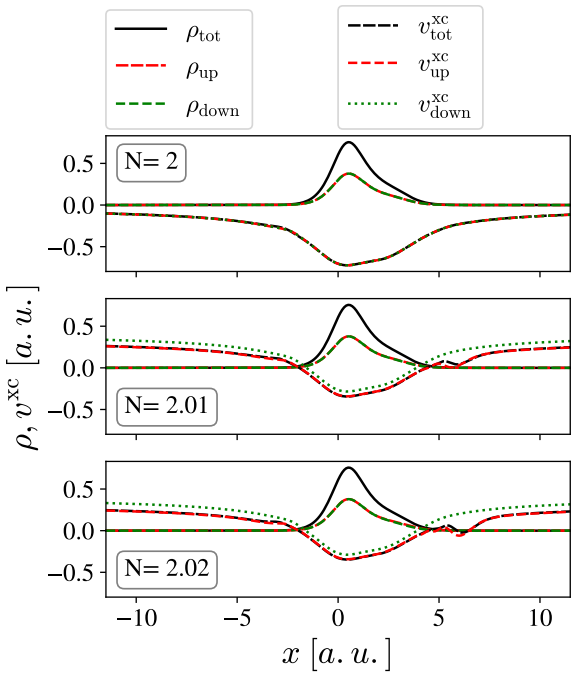

FIG. 7. Densities and corresponding xc potentials for a certain external potential for different fractional particle numbers.

For the ground state of $N = 3$ two linear independent spin-eigenfunctions can be chosen:

$$X\,(3, 1/2, 1/2; 1) = \frac{1}{\sqrt{6}}\left[2\uparrow\uparrow\downarrow - (\uparrow\downarrow\uparrow + \downarrow\uparrow\uparrow)\right], \tag{11a}$$

$$X\,(3, 1/2, 1/2; 2) = \frac{1}{\sqrt{2}}\left(\uparrow\downarrow\uparrow - \downarrow\uparrow\uparrow\right). \tag{11b}$$

The (normalized) spatial parts in Eq. (9) and the corresponding spin densities $\rho_\uparrow(x)$ and $\rho_\downarrow(x)$ can then be found. For instance,

$$\rho_\uparrow(x) = \iint \left[5|\tilde{\Phi}_1|^2 + 9|\tilde{\Phi}_2|^2\right](\mathrm{d}x_2\mathrm{d}x_3 + \mathrm{d}x_1\mathrm{d}x_3) + \iint \left[2|\tilde{\Phi}_1|^2 + 18|\tilde{\Phi}_2|^2\right]\mathrm{d}x_1\mathrm{d}x_2\,, \tag{12}$$

with $\tilde{\Phi}_1 = 2\,\Phi_{11}^{S=1/2}$ and $\tilde{\Phi}_2 = 2\Phi_{21}^{S=1/2}/\sqrt{3}$.

Kohn-Sham inversion

For the inversion of the densities we used the optimization algorithm proposed in Ref. [44], that casts the inverse DFT problem of finding the $v_{\mathrm{xc}}(\boldsymbol{r})$ that yields a given density $\rho(\boldsymbol{r})$ as a constrained optimization problem. The conjugate gradient method was used to update the xc potentials $v^\sigma_{\mathrm{xc}}(\boldsymbol{r})$. A constant weight function $w \equiv 1$ was also used.

The densities of $N = 1, 2, 3$ particles were mixed, allowing us to generate a fractional densities for each external potential. For instance, the set $\{1, 1.5, 2, 2.5, 3\}$ contains additional $\epsilon = 0.5$ fractional densities beside the integer ones. If the inversion algorithm did not converge to a MSE below 1.5×10^{-7} for a given density, we removed all samples corresponding to that external potential. The computed xc potentials were shifted by a constant to be in agreement with Koopman's theorem [60].

In Fig. 6 we show a few examples of such spin-densities and inverted xc potentials. For the fractional densities plotted in Fig. 7 the shift of the xc potential caused by the Δ_{xc} jump is clearly visible.

Network implementation

The neural networks were implemented in pytorch [61] using pytorch-lightning [62] to simplify the training process. As discussed before we used the sliding window convolution as proposed in Ref. [13]. For each network the number of zero-paddings at the system's boundaries was set to $(\kappa - 1)/2$ (κ is an odd number in our calculations). For all models presented in this paper, we chose a kernel size of $\kappa = 201$, corresponding to highly nonlocal funtionals. Each local density was fed into a fully connected network with SILU [63] activation functions, and the output layer returned the *local* functions described in the previous sections. The use of SILU activation functions guaranteed the smoothness of the exchange correlation potential and its higher order derivatives. All hidden layer sizes were set to 32. Including the window convolu-

tion we used 4 hidden layers. The additional SWC unit for the AF shared the same hyperparameters. The network weights were optimized with ADAM while using a cyclic learning rate scheduler. We used a learning rate of 7×10^{-4} with a batch size of 30. The only exceptions were models trained with the xc-jump in the loss function. In this case each batch contained one density, corresponding to a random external potential. If the Δ_{xc}-shift was incorporated in the loss function, the densities per batch consisted of *all* fractional densities of a certain external potential. For these networks we kept the learning rate used before, but changed the batch size to one. The models were trained for 30 000 epochs and the model with the best validation loss was selected for testing.

[1] P. Hohenberg and W. Kohn, Phys. Rev. **136**, B864 (1964).
[2] R. O. Jones, Rev. Mod. Phys. **87**, 897 (2015).
[3] W. Kohn and L. Sham, Phys. Rev. **140**, A1133 (1965).
[4] S. Lehtola, C. Steigemann, M. J. Oliveira, and M. A. Marques, SoftwareX **7**, 1 (2018).
[5] J. Perdew and K. Schmidt, AIP Conf. Proc. **577**, 1 (2001).
[6] L. Deng and X. Li, IEEE-ACM T. Audio Spe. **21**, 1060 (2013).
[7] B. Kalita, L. Li, R. J. McCarty, and K. Burke, Acc. Chem. Res. **54**, 818 (2021).
[8] F. Brockherde, L. Vogt, L. Li, M. Tuckerman, K. Burke, and K.-R. Müller, Nat. Comm. **8**, 872 (2017).
[9] L. Li, S. Hoyer, R. Pederson, R. Sun, E. Cubuk, P. Riley, and K. Burke, Phys. Rev. Lett. **126**, 036401 (2021).
[10] J. Margraf and K. Reuter, Nature Comm. **12**, 344 (2021).
[11] J. R. Moreno, G. Carleo, and A. Georges, Phys. Rev. Lett. **125**, 076402 (2020).
[12] K. Ryczko, D. A. Strubbe, and I. Tamblyn, Phys. Rev. A **100**, 022512 (2019).
[13] J. Schmidt, C. L. Benavides-Riveros, and M. A. L. Marques, J. Phys. Chem. Lett. **10**, 6425 (2019).
[14] R. Nagai, R. Akashi, and O. Sugino, npj Comput. Mater. **6**, 43 (2020).
[15] M. M. Denner, M. H. Fischer, and T. Neupert, Phys. Rev. Research **2**, 033388 (2020).
[16] E. Lieb, Int. J. Quantum Chem. **24**, 243 (1983).
[17] E. Ludeña and V. Karasiev, Kinetic energy functionals: History, challenges and prospects, in *Reviews of Modern Quantum Chemistry*, edited by K. Sen (World Scientific, 2002) pp. 612–665.
[18] E. Kraisler and L. Kronik, Phys. Rev. Lett. **110**, 126403 (2013).
[19] L. J. Sham and M. Schlüter, Phys. Rev. Lett. **51**, 1888 (1983).
[20] J. Perdew, R. Parr, M. Levy, and J. Balduz, Phys. Rev. Lett. **49**, 1691 (1982).
[21] M. Grüning, A. Marini, and A. Rubio, Phys. Rev. B **74**, 161103 (2006).
[22] X. Andrade and A. Aspuru-Guzik, Phys. Rev. Lett. **107**, 183002 (2011).
[23] M. J. P. Hodgson, E. Kraisler, A. Schild, and E. K. U. Gross, J. Phys. Chem. Lett. **8**, 5974 (2017).
[24] A. Mirtschink, M. Seidl, and P. Gori-Giorgi, Phys. Rev. Lett. **111**, 126402 (2013).
[25] P. Mori-Sánchez and A. Cohen, Phys. Chem. Chem. Phys. **16**, 14378 (2014).
[26] M. Mosquera and A. Wasserman, Mol. Phys. **112**, 2997 (2014).
[27] E. J. Baerends, Mol. Phys. **118**, e1612955 (2020).
[28] J. Perdew, in *Density Functional Methods In Physics*, edited by R. Dreizler and J. d. Providência (Plenum Press, 1985) pp. 265–308.
[29] A. J. Cohen, P. Mori-Sánchez, and W. Yang, Chem. Rev. **112**, 289 (2012).
[30] W. Yang, Y. Zhang, and P. Ayers, Phys. Rev. Lett. **84**, 5172 (2000).
[31] J. Perdew and M. Levy, Phys. Rev. Lett. **51**, 1884 (1983).
[32] F. Eich and M. Hellgren, J. Chem. Phys. **141**, 224107 (2014).
[33] S. Kümmel and L. Kronik, Rev. Mod. Phys. **80**, 3 (2008).
[34] A. Ruzsinszky, J. Perdew, G. Csonka, O. Vydrov, and G. Scuseria, J. Chem. Phys. **125**, 194112 (2006).
[35] E. Perfetto and G. Stefanucci, Phys. Rev. B **86**, 081409 (2012).
[36] C. Li and W. Yang, J. Chem. Phys. **146**, 074107 (2017).
[37] J. C. Slater and K. H. Johnson, Phys. Rev. B **5**, 844 (1972).
[38] L. G. Ferreira, M. Marques, and L. K. Teles, Phys. Rev. B **78**, 125116 (2008).
[39] E. J. Baerends, J. Chem. Phys. **149**, 054105 (2018).
[40] D. Hait and M. Head-Gordon, J. Phys. Chem. Lett. **9**, 6280 (2018).
[41] A. Cohen, P. Mori-Sánchez, and W. Yang, Science **321**, 792 (2008).
[42] P. Mori-Sánchez, A. J. Cohen, and W. Yang, Phys. Rev. Lett. **100**, 146401 (2008).
[43] C. L. Benavides-Riveros, N. N. Lathiotakis, and M. A. L. Marques, Phys. Chem. Chem. Phys. **19**, 12655 (2017).
[44] B. Kanungo, P. M. Zimmerman, and V. Gavini, Nature Comm. **10**, 4497 (2019).
[45] U. von Barth and L. Hedin, J. Phys. C **5**, 1629 (1972).
[46] M. Hellgren and E. K. U. Gross, Phys. Rev. A **85**, 022514 (2012).
[47] V. Nair and G. Hinton (2010) pp. 807–814.
[48] M. Medvedev, I. Bushmarinov, J. Sun, J. Perdew, and K. Lyssenko, Science **355**, 49 (2017).
[49] B. Senjean and E. Fromager, Phys. Rev. A **98**, 022513 (2018).
[50] P.-F. Loos and E. Fromager, J. Chem. Phys. **152**, 214101 (2020).
[51] C. L. Benavides-Riveros, J. Wolff, M. A. L. Marques, and C. Schilling, Phys. Rev. Lett. **124**, 180603 (2020).
[52] E. Kraisler and A. Schild, Phys. Rev. Research **2**, 013159 (2020).
[53] J. Cioslowski, J. Chem. Phys. **153**, 154108 (2020).
[54] X. Andrade *et al.*, Phys. Chem. Chem. Phys. **17**, 31371 (2015).
[55] R. Pauncz, *The Construction of Spin Eigenfunctions* (CRC Press, 2000).
[56] J. Whitfield, J. Chem. Phys. **139**, 021105 (2013).
[57] W. Greiner, *Quantum Mechanics. Symmetries* (Springer, 1994).
[58] R. McWeeny, *Methods of molecular quantum mechanics* (Academic Press, 1992).

[59] F. Porter, Group Theory (2009).
[60] O. Gritsenko and E. Baerends, J. Chem. Phys. **117**, 9154 (2002).
[61] A. Paszke *et al.*, in *Advances in Neural Information Processing Systems 32* (Curran Associates, Inc., 2019) pp. 8024–8035.
[62] W. A. Falcon *et al.*, GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning **3** (2019).
[63] D. Hendrycks and K. Gimpel, Gaussian error linear units (gelus) (2020), arXiv:1606.08415.

COMPETING INTERESTS

The authors declare no competing interests.