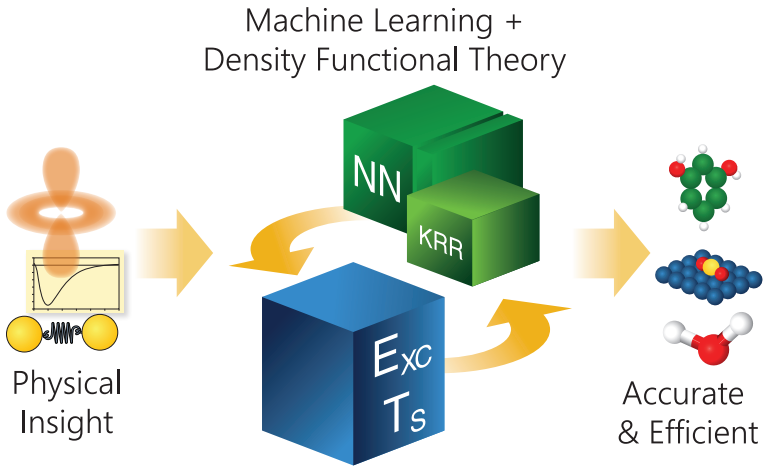
Learning to Approximate Density Functionals

Bhupalee Kalita,[1] Li Li,[2] Ryan J. McCarty,[1] and Kieron Burke[1,3]

[1] *Department of Chemistry, University of California, Irvine, CA 92697*
[2] *Google Research, Mountain View, CA 94043*
[3] *Department of Physics and Astronomy, University of California, Irvine, CA 92697*[*]

Density functional theory (DFT) calculations are used in over 40,000 scientific papers each year, in chemistry, materials science, and far beyond. DFT is extremely useful because it is computationally much less expensive than *ab-initio* electronic structure methods and allows systems of considerably larger size to be treated. But the accuracy of any Kohn-Sham DFT calculation is limited by the approximation chosen for the exchange-correlation (XC) energy. For more than half a century, humans have developed the art of such approximations, using general principles, empirical data, or a combination of both, typically yielding useful results, but with errors well above the chemical accuracy limit (1 kcal/mol). Over the last 15 years, machine learning (ML) has made major breakthroughs in many applications and is now being applied to electronic structure calculations. This recent rise of ML begs the question: Can ML propose or improve density functional approximations? Success could greatly enhance the accuracy and usefulness of DFT calculations without increasing the cost.

In this work, we detail efforts in this direction, beginning with an elementary proof of principle from 2012, namely finding the kinetic energy of several fermions in a box using kernel ridge regression. This is an example of orbital-free DFT, for which a successful general-purpose scheme could make even DFT calculations run much faster. We trace the development of that work to state-of-the-art molecular dynamics simulations of resorcinol with chemical accuracy. By training on *ab-initio* examples, one bypasses the need to find the XC functional explicitly. We also discuss how the exchange-correlation energy itself can be modeled with such methods, especially for strongly correlated materials. Finally, we show how deep neural networks with differentiable programming can be used to construct accurate density functionals from very few data points by using the Kohn-Sham equations themselves as a regularizer. All these cases show that ML can create approximations of greater accuracy than humans, and is capable of finding approximations that can deal with difficult cases such as strong correlation. But such ML-designed functionals have not been implemented in standard codes because of one last great challenge: generalization. We discuss how effortlessly human-designed functionals can be applied to a wide range of situations, and how difficult that is for ML.



1. INTRODUCTION

Direct solution of the Schrödinger equation for electrons (traditionally designated as *ab initio* in quantum chemistry) yields chemically accurate energies (errors below 1 kcal/mol). However computational costs scale poorly with system size, limiting its routine applicability to smaller molecules. On the other hand, density functional theory (DFT) calculations typically scale much more favorably, allowing routine calculation of molecules with hundreds of atoms. This increased applicability comes at a cost: The effective non-interacting Kohn-Sham equations that, in principle, yield exact ground-state energies and densities, in practice, require a small fraction of the total energy (called the exchange-correlation (XC) energy) to be approximated

* kieron@uci.edu

in an uncontrolled way.

Presently, there are hundreds of distinct approximations to the XC energy [1], all of which are available in common electronic-structure codes. Some have been designed from general principles of physics, without reference to any specific molecular or material system [2]. Others have been fitted and tested on an ever-growing population of databases of distinct molecular systems and properties and these yield higher accuracies on those systems [3]. However, almost all use similar basic ingredients, such as the density, its gradient, and a fraction of Hartree-Fock (HF) exchange, and are inspired by physical or chemical insight.

In the past decade, machine learning (ML) has seen some remarkable successes in various applications, including image recognition, language translation [4], and even playing curling [5]. ML is also increasingly being applied to problems in physical sciences, where it can help with, for example, extraction of salient features from microscopy images [6], or climate modelling [7]. It can also be used to speed up purely computational tasks. In electronic structure theory, there has been much success in designing new force-fields using ML, creating far more accurate force fields than previous human-designed attempts [8]. ML force-fields can reproduce results from DFT or any *ab-initio* methods at a fraction of the computational cost, simply by training on carefully chosen examples, and are already available in useful codes [9].

A different, and arguably more difficult, task is to use ML to design new density functional approximations or to improve existing ones. This is simply a regression problem, i.e., fitting a function of many variables. But regression in DFT involves fitting a functional, which can be considered a function of infinitely many variables and that complicates the task.

There are several distinct approaches to using ML to make functionals. If the goal is to make DFT calculations run faster, one such problem is approximating the KS kinetic energy functional, i.e., the kinetic energy of the non-interacting KS orbitals ($T_{\rm S}[n]$), thereby bypassing the need to solve the KS equations, the most expensive step in most DFT implementations. If $T_{\rm S}$ could be computed rapidly, it could revolutionize all DFT calculations by making them run much faster [10]. This is called orbital-free DFT (OF-DFT) [11–14]. Moreover, training data is abundant as every self-consistent cycle of every DFT calculation ever performed yields a set of orbitals (and hence density) and their $T_{\rm S}$. But the path to success is not smooth. To determine the density in OF-DFT, one must solve an Euler equation [15] requiring an accurate and well-behaved functional derivative of $T_{\rm S}$. Due to limited information available in direct training, ML-designed interpolating functionals that are extremely accurate for the energy, almost necessarily yield poor functional derivatives.

The more traditional problem is to improve the accuracy of DFT, either by modifying existing XC approximations or creating completely new forms [16–18]. Usually (but not always [19]) the functional derivative of the XC energy is somewhat unimportant to the energy. But unlike the orbital-free approach, the amount of accessible accurate training data from higher level of theories is limited and is mostly available for relatively small systems. Nonetheless, promising ML ideas developed for OF-DFT can also be applied to the XC case. Combining both can improve accuracy and computational cost simultaneously[20].

Another important objective is to find new forms that overcome the drawbacks of traditional human-designed XC approximations. For instance, most molecules and many materials in their equilibrium state are considered to be weakly correlated, where ingredients that have been used in the past work reasonably well and can be borrowed to design ML functionals too. But most XC approximations fail to break bonds correctly because they fail when a bond is stretched and electrons localize on distinct sites. Thus the complete binding energy curves of even H_2^+ and H_2 represent paradigm difficult problems for standard DFT [21]. A stretched bond is an example of strong correlation that provides a good test for ML-designed functionals. Fig. 1 shows an ML-functional reproducing an entire binding energy curve from training on only two bond lengths [18]. Such bonds are even more difficult for OF-DFT if semi-local approximations (terms that depend on only the density and its gradient) [2, 22] are used, as the same considerations apply even more strongly to $T_{\rm S}[n]$.

In principle, ML-designed functionals need not be limited by human imagination and intuition as ML can use the density everywhere to find the energy contribution at a point (a fully non-local functional) [13]. This is an ambitious goal. Humans have an almost 100-year head-start on this task [1], and it may be a while before an ML functional become as useful and practical as B3LYP[22] in chemistry. In current studies, many simplifications are made for efficient data generation and easier implementation, simply to see if a new ML approach *can* work, before building more realistic or general applications.

Thus, several of the examples discussed here are for one-dimensional analogs of true electronic structure problems [11–13, 18, 20]. For the non-interacting problem, an effective code can be written in a few minutes for solving the Schrödinger equation, and training data generated within hours on a single core. For interacting systems, highly accurate solutions can be obtained very efficiently in one dimension, using a method called the density matrix renormalization group (DMRG) [23]. DMRG is a very powerful quantum solver, using matrix product states, with many applications to strongly correlated model systems relevant to condensed matter physics [24] and also in quantum chemistry [25]. Recently, considerable effort was made to create a one-dimensional analog of molecular systems using DMRG to handle strongly correlated effects [26], making data generation much easier. Such simplicity ensures maximum flexibility and ease in interfacing with existing ML codes, which often come in prepackaged routines.

A useful introduction to ML for chemical scientists can be found in Ref. [27] with a glossary of terms. Here, we simply distinguish between kernel methods and deep neural

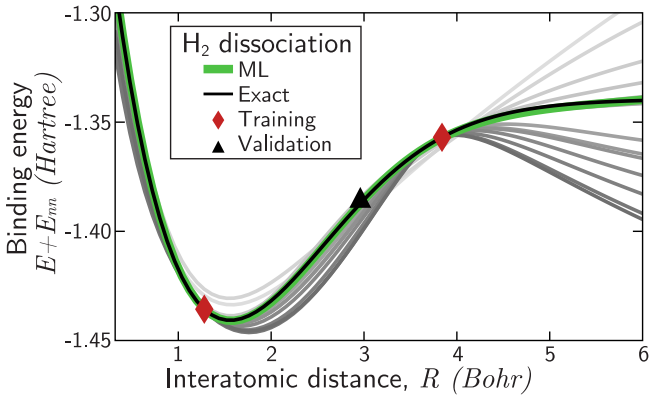


FIG. 1. The dissociation curve of a one-dimensional H_2 molecule, created using the ML XC approximation of Ref. 18 by training with DMRG data at just two configurations. Darkening shades of grey show predictions from underfitting to overfitting, but distributed around the exact curve due to the physics prior knowledge built into the model. The optimal green curve, found by validating the model at a single configuration, produces chemically accurate results. $E_{\rm nn}$ is the nucleus-nucleus repulsion energy. See Fig. 11 for details.

networks, the two methods used in the key references. The basic problem is one of regression with many parameters, where some method of regularization is required to avoid overfitting. Regularization is any procedure that allows one to control how smooth the fit is. Ridge regularization penalizes overfitting with the sum of the squares of the fitting coefficients. The kernel trick maps a low-dimensional space to a higher one to create a function that is easier to fit [28], which is especially relevant in our case. Kernel ridge regression (KRR) remains a standard tool in ML today.

However, many of the most impressive gains in ML have recently come from neural networks (NN). These are characterized by the graph of differentiable operations, architectures with various inductive biases, and scalability on hardware accelerators [29]. Their performance can usually be continuously improved by increasing the model capacity, with copious addition of data, whereas more traditional methods can saturate or become too expensive to train [30]. In fact, the first application of ML to density functional design was using NN [31]. This pioneering work used exact energies and XC potentials to fit an XC functional that remains relevant even today. In this article, we discuss the chronological developments of ML density functionals focusing only on the work of our research group, but comprehensive reviews are available elsewhere [32].

2. PROTOTYPE

Here we review the most elementary application of ML to create an approximate OF-DFT functional[11]. The simplest problem imaginable is to consider the energy levels of a 1D potential between infinite walls. It is trivial to solve such box problems numerically, filling the levels with same-spin fermions, so that there is one particle per level. For N fermions in the box, the KRR kinetic energy functional is:

$$T^{\rm ML}[n] = \sum_{j=1}^{N_T} \alpha_j k(n, n_j), \tag{1}$$

where N_T is the number of training densities, α_j are the weights, and k is a Gaussian kernel of the form

$$k(n, n_j) = \exp(-\int d^3r (n(\mathbf{r}) - n_j(\mathbf{r}))^2 / 2\sigma^2). \tag{2}$$

The weights α_j are found by minimizing the mean squared error of $T^{\rm ML}[n]$ for all training data plus a regularization penalty, while σ can be determined by cross-validation. Each data point adds an integral over the entire density inside the Gaussian kernel, and hence the resulting functional is completely non-local.

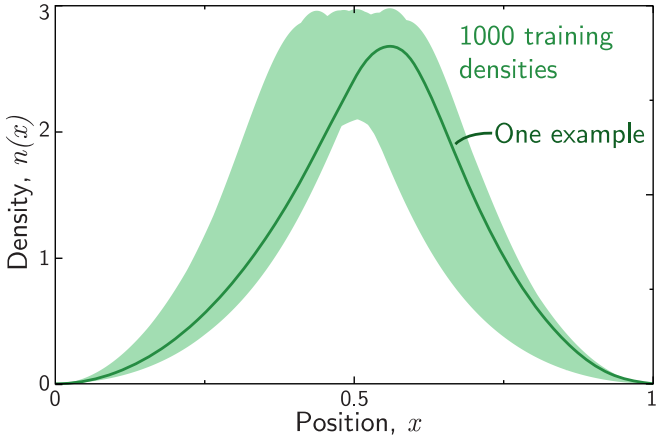


FIG. 2. The range of variation within the data set of 1000 training densities for $N = 1$ for the box problem (green). Any one of these densities can be accurately reproduced using the projection method discussed in Ref. 11. Adapted with permission from Ref. 11. Copyright 2012 American Physical Society.

To generate data, three Gaussian potential dips were placed at random inside the box. For $N = 1$, with as few as 80 training densities, chemically accurate (error less than 1 kcal/mol) predictions were made for the kinetic energies of a test set drawn from the same distribution, and shown in Fig. 2. This was a huge improvement compared to semi-local XC approximations (error = 160 kcal/mol). However, to be useful, an approximate $T_{\rm S}$ must also have an accurate derivative, so that the Euler equation yields an accurate density [15]. The functional derivative of KRR $T^{\rm ML}[n]$ has the form,

$$\frac{\delta T^{\rm ML}}{\delta n(x)} = \sigma^{-2} \sum_{j=1}^{N_T} \alpha_j \left(n_j(x) - n(x) k(n, n_j)\right). \tag{3}$$

This derivative is shown in Fig. 3. It oscillates wildly relative to the exact curve. This is expected as the exact functional

derivative describes the change in the functional in every direction in the infinite-dimensional space of densities, but with KRR, one could only expect it to be accurate in the very few directions in which it has training data.

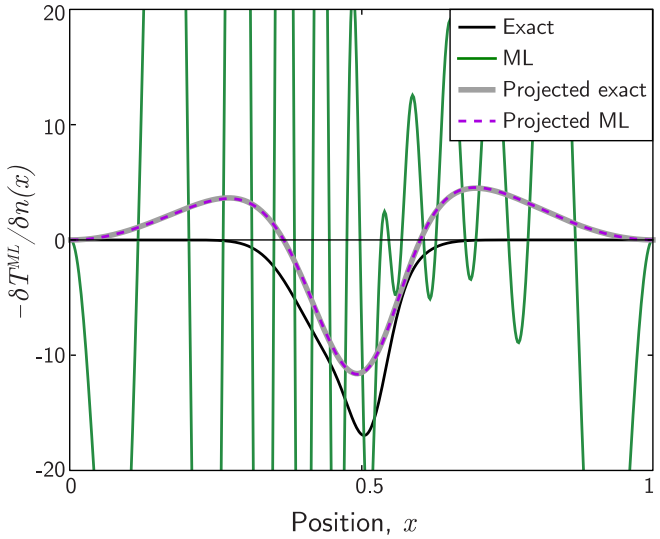


FIG. 3. Functional derivative of $-T^{\mathrm{ML}}[n]$, the exact derivative, $v(x)$, and their projections on the data-manifold for $N_T = 100$. Adapted with permission from Ref. 13. Copyright 2015 John Wiley and Sons.

To overcome this problem, a constraint was added to the minimization, $\delta(E[n] - \zeta g[n]) = 0$, where the functional $g[n] = 0$ defines the manifold of training densities. The specific $g[n]$ can be determined using principal component analysis (PCA) [27]. The cartoon in Fig. 4 illustrates this process. One first calculates the usual functional derivative, and then projects it onto the local principal components in which there are greatest variations among the nearby training densities. This leads downhill on the training manifold, and since the optimal density should be within that manifold, it finds a density very close to the exact minimizer. Although the projected derivative is very accurate, as in Fig. 3, the error of the functional evaluated on this projected ML density, $T^{\mathrm{ML}}[n^{\mathrm{ML}}]$, is substantially larger than that of T^{ML} on the exact density, chemical accuracy is still achieved with 150 training samples for one particle.

A detailed account of all the KRR implementation is given in Li et al.[13]. Six alternative kernels were tried, of which three had comparable performance, including the Gaussian used here. The details of how the projection method works are also explained, discussing the relative contributions of the energy and density to the error. An analysis of the functional found, and the hyperparameter landscape, is available in Ref. 33.

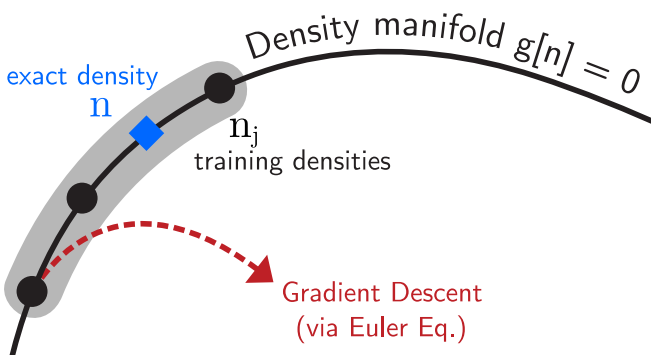


FIG. 4. The training densities and the exact density are on the density manifold defined by $g[n] = 0$. The solution of the Euler equation via simple gradient descent becomes unstable (red dashed curve) and leaves the shaded region.

3. ORBITAL-FREE DFT

Inspired by the proof of principle from Ref. 11, many questions arise as one works towards chemical realism.

A. Bond breaking

Orbital-free semi-local approximations to $T_{\mathrm{S}}[n]$ fail worse than those for XC when a chemical bond is stretched. An implementation of KRR to correctly describe the stretched bond limit can be found in Snyder et al.[12]. They trained $T_{\mathrm{S}}^{\mathrm{ML}}[n]$ with data from KS-DFT along the bond distance of several prototype 1D diatomic molecules and tested if the non-local ML approximation, similar to the one in the box problem, could remain accurate all along the dissociation curve. To tackle the highly curved density manifold, a technique called nonlinear gradient denoising (NLGD) was also proposed. By utilizing kernel principal component analysis (kPCA) [34] to capture the low-dimensionality, this method improves the accuracy of the projected gradient descent with even fewer training densities compared to normal PCA in Ref. 11.

For both H_2 and LiH, the relative error in $T_{\mathrm{S}}^{\mathrm{ML}}[n]$ evaluated on the projected density with NLGD was less than 1 kcal/mol with just $N_T = 10$. By increasing the training set size to 20, the bond dissociation energy, equilibrium bond length, and the zero-point vibrational frequency could be determined to within 1%. Fig. 5 depicts how accurately the ML algorithm reproduces the exact binding energy curve of H_2 obtained from a DFT calculation.

The NLGD algorithm is further illustrated in Ref. 35 for the 1D box problem. A 3D expansion of a similar OF-DFT mapping can be found in Ref. 36 where a convolutional neural network predicts the potential energy surface for hydrocarbon chains with accuracy comparable to those of human-designed functionals. Examples of improvements made in human-designed functionals for the same problem can be found in Seino et al.[37] and Golub et al.[38], who

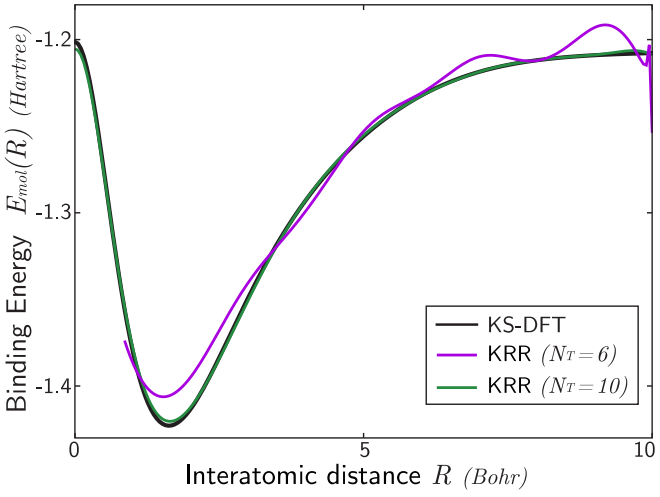


FIG. 5. The molecular binding energy curve obtained with constrained optimal densities (KRR-NLGD) for 1D model of H_2. Adapted with permission from Ref. 12. Copyright 2013 AIP Publishing.

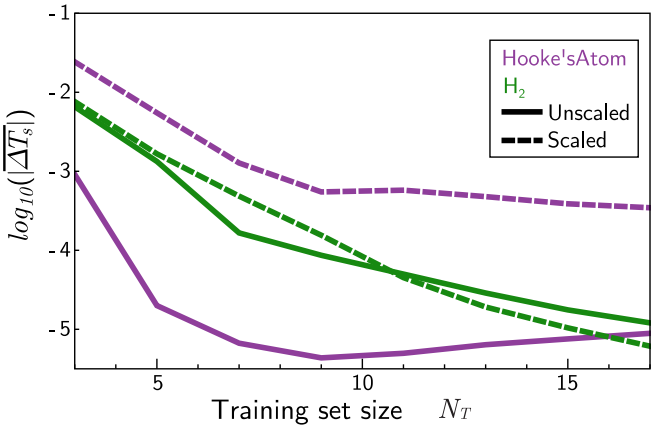


FIG. 6. The error in the kinetic energy functional trained on scaled and unscaled densities for both 1D Hooke's atom and 1D H_2 molecule. Adapted with permission from Ref. 40. Copyright 2018 AIP Publishing.

trained neural networks for $T^{\mathrm{ML}}[n]$ that included up to third-order and fourth-order gradients of the density.

B. Exact conditions

In DFT, known theoretical properties (exact conditions) are used to constrain the form of approximate functionals[2, 22, 39]. But the ML models above cannot be analyzed by checking for such conditions. The weights in the KRR functional are large and alternate in sign, suggesting the possibility of predicting totally unphysical negative kinetic energy. However, all test densities considered had accurate positive ML kinetic energies, i.e., throughout the training density manifold.

In order to make these KRR functionals less system-specific and to enable easier training, a later study[40] incorporated one of the elementary exact conditions of DFT, the coordinate scaling, within the KRR optimization,

$$T_{\mathrm{S}}[n_\gamma] = \gamma^2 T_{\mathrm{S}}[n],\ n_\gamma(\mathbf{r}) = \gamma^3 n(\gamma\mathbf{r}),\ \gamma > 0. \qquad (4)$$

Two 1D systems were studied separately- the exactly solvable Hooke's atom, and the H_2 molecule with accurate DMRG energies and densities. After training the KRR model on scaled density n_γ, it was evaluated on a test set of 50 densities for the two systems. Fig. 6 shows that in Hooke's atom the scaled kinetic energy functional was much more accurate than its unscaled counterpart, but not for H_2.

Scaling makes the densities of different configurations of Hooke's atom look similar to one another. But that is not so for H_2, hence no improvement is seen in its kinetic energy. This is a result of the large changes in density as you move within the training manifold. Would scaling improve learning if several molecules at different bond distances were simultaneously trained on?

C. Molecular dynamics of single molecules

New complications arise when ML is applied to chemically realistic problems. Brockherde et al.[14] tried incorporating these methods in realistic 3D electronic structure codes, but as the number of degrees of freedom increased, the cost of the projection method to determine the density became prohibitive. A relatively simple workaround is to learn the density directly as a functional of the potential and so bypass the need to solve either the KS equations or the Euler equation. The KRR density and energy models in Ref. 14 were capable of running molecular dynamics (MD) with a standard XC approximation (PBE) for a small organic molecule, malonaldehyde. Training sets were generated by running classical MD simulations at higher temperatures, e.g. 500 K (to ensure sampling of higher energy regions of the potential energy surface) and then performing DFT calculations at snapshots of such simulations. With sufficient training, the errors in the density map became much smaller than density differences due to different XC approximations.

The performance of this ML density functional along the MD trajectory is shown in Fig. 7. In the region where the proton transfer occurs, the error is largest because these configurations are not included in the training set. One could easily run a KS calculation for this particular configuration and retrain including that data point to reduce this error. In fact, standard KS-MD does not yield accurate proton transfer rates, as nuclear tunneling plays an important role and requires more sophisticated approaches[41].

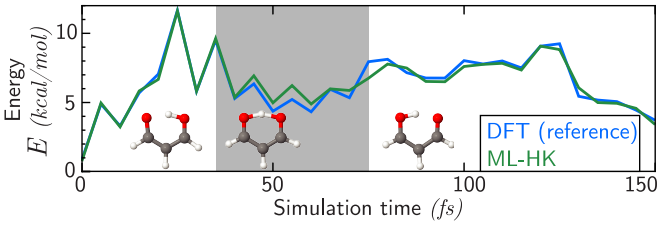


FIG. 7. Predicted ML energy along a 0.15 ps MD trajectory of malonaldehyde showing the transfer of a proton between oxygen atoms. Adapted from Fig. 5 of Ref. 14. Copyright 2017 Nature Research, licensed under Creative Commons Attribution 4.0 International.

D. Δ-DFT and chemical accuracy

Although the training data used for malonaldehyde were generated from approximate DFT, in principle, the ML functional could also be trained on energies and densities from higher-level *ab-initio* theories, such as coupled-cluster, i.e. to bypass the KS equations, as if they had been solved with chemical accuracy.

In practice, it is difficult to extract accurate densities for training from a CCSD(T) calculation[42], but one can simply learn accurate energies as a functional of the density of a standard DFT calculation. This leads to several different energy functionals that ML can produce: the *ab-initio* energy, the DFT energy, and the difference in the two (Δ-DFT), which is much easier to learn (i.e., converges much more rapidly with training data) because the error in a DFT calculation is a very smooth function of the nuclear coordinates. All this was done in a recent work by Bogojeski, Vogt-Maranto, et al.[43]. Of many different situations studied, the highlight is again ML-MD simulations, in which a rotation barrier in resorcinol was probed. A semi-local XC functional makes a substantial error in the rotation barrier and Fig. 8 shows how the DFT trajectory bifurcates from the accurate trajectory. The KRR-DFT energy on the ML density yields almost perfect agreement with a full DFT MD simulation. Self-consistent DFT corrected with Δ-DFT calculated on the ML density yields trajectories with errors less than 0.2 kcal/mol. Using the ML density with the CCSD(T) energy without performing DFT calculations at each step, usually gives a good trajectory, but with substantial energy errors. Moreover, directions can appear in a trajectory that are wholly unphysical, taking the molecule outside the manifold on which the density functional works.

Unfortunately, it is difficult to generalize these methods to other systems or to strong correlation. A similar machine-learned correcting functional was also defined in Dick et al.[44] for liquid water which used an NN to predict accurate ground-state properties by approximating the difference in energies and forces from the DFT densities. Later, an approximation for XC was also constructed with this method[17].

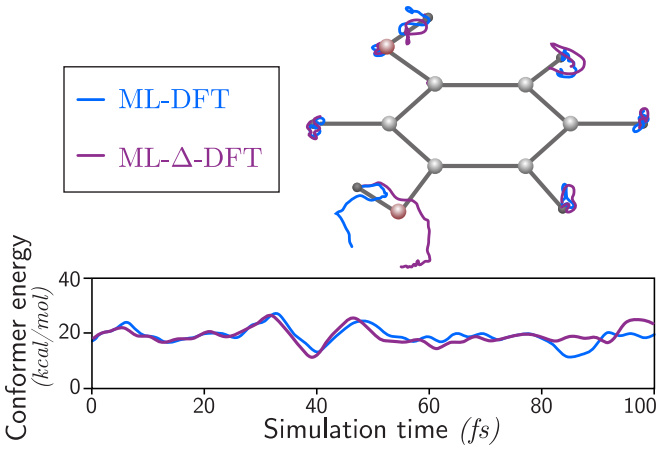


FIG. 8. Positions and energy of the resorcinol conformer switch predicted using standard DFT alone (blue), and after correction with Δ−DFT trained on CCSD(T) energies (purple). Adapted from Fig. 3 of Ref. 43. Copyright 2020 Nature Research, licensed under Creative Commons Attribution 4.0 International.

4. EXCHANGE-CORRELATION:

We turn now to models for XC. Much work in the literature applies to weakly correlated systems. We focus on creating fully non-local ML approximations so that strong correlation can be handled. Because highly accurate densities and energies are cumbersome and expensive to generate for training, we return to the simpler 1D world for testing these ideas.

A. Strong correlation and thermodynamic limit

For materials applications, true strong correlation is even worse than in stretched H_2. For example, for stretched H_4, semi-local XC approximations create four broken spin-symmetry solutions, not two. Ultimately, for solid-state applications, one should be able to handle the infinite-chain, or in other words, the thermodynamic limit[26].

In Li et al. [20], the task was to learn both T_s and XC and their derivatives for 1D H-atom chains of fixed separation varying from equilibrium to very stretched, and chains varying from two to twenty atoms, to accurately extrapolate to the thermodynamic limit. Contrary to Ref. 12, the KRR machinery was applied to DMRG energies and densities to approximate both $T_s[n]$ and $E_{xc}[n]$ in one shot. This was an extremely ambitious goal given the requirement of accurate functional derivatives and the enormous size of the kinetic and Hartree energies. The NLGD method described in the previous sections[12] yields an extremely accurate 1D H_2 dissociation curve. But this method becomes far too costly for longer chains as the number of grid points in the density increases. Without accurate derivatives, one can still easily learn energies, but not calculate accurate densities.

The key was the representation of the density. There

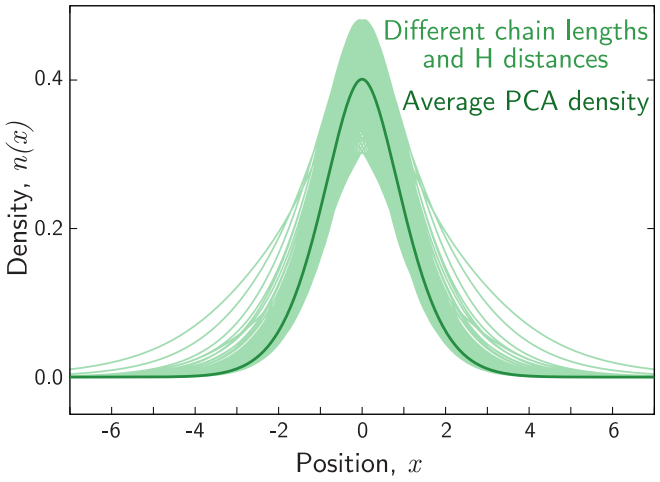


FIG. 9. Individual Hydrogen partition densities for every interatomic separation R within the training set for a chain of length N and the base density found using PCA. Note similarity to Fig.2. Adapted with permission from Ref. 20. Copyright 2016 American Physical Society.

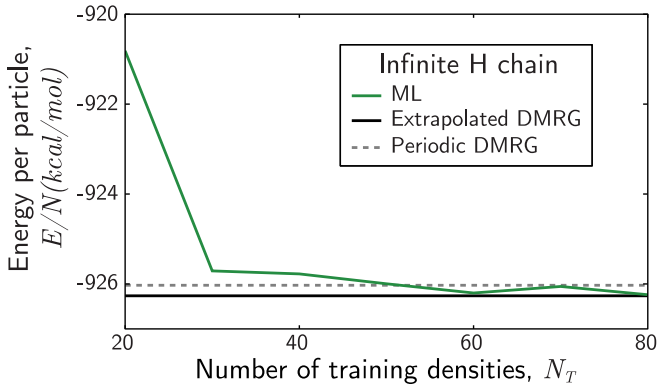


FIG. 10. Energy of the infinite H-chain with uniform interatomic spacing of 2.08 Bohr trained using extrapolated DMRG chain densities and energies. Adapted with permission from Ref. 20. Copyright 2016 American Physical Society.

is too much freedom when it is simply a function of the large grids needed to represent the system. Many alternative representations were tried, but the ultimate winner was the simple atoms-in-molecules partitioning of Hirshfeld [45]. A molecular density of an N-atom chain was decomposed into a weighted sum of distorted atomic densities. After collecting and centering all these atomic densities, PCA was used to create a data-driven basis for the allowed density variations shown in Fig. 9. This reduced the time needed to calculate the optimizing densities by several orders of magnitude while retaining chemical accuracy. The infinite-chain limit of 1D H-atoms could then be found with chemical accuracy, treating all aspects of the DFT calculation with KRR on a PCA basis learned from atoms-in-molecules. DMRG results for both extrapolation of finite chains and periodic systems agreed with each other and with the ML result to within 1 kcal/mol (Fig. 10).

On reflection, it would have been much easier to simply approximate the XC energy alone with ML methods in this calculation, and use the KS procedure to produce accurate densities. This seems a worthwhile test for future work, and might also have been useful in Ref. 12.

Other studies have also tried to address strong-correlation with ML-DFT on model systems[46–48]. But developments are more prominent for weakly correlated systems[49–51].

B. Kohn-Sham regularizer (KSR)

Here, we look again at full binding energy curves obtained with DMRG to find XC approximations that correctly break bonds, but now within the KS framework. A pioneering study [16] showed that by including density errors in the loss function of a feed-forward NN, one could achieve performance comparable to human-designed functionals for a real molecule by training on just three or four molecules. This is because the density is the functional derivative of the energy with respect to the external potential, by training with densities one simultaneously improves both the energy and all possible linear responses to changes in the potential. This greatly enhances the possibilities of generalization.

There are several other efforts to build a transferable ML-DFT model with different approaches[48, 52–54]. The most recent work by Li et al.[18] pushes the inspiration from Ref. 16 forward in two major respects. The first is to see if an entire dissociation curve can be found with minimal training on a few examples. The second is a theme of deep learning in general, namely the importance of differentiable programming (DP). DP keeps rigorous components where we have paramount physics prior knowledge and well-established numerical methods. By using DP, one can automatically apply gradient-based approaches to optimization, unlike earlier work.

NNs often have many more parameters than training examples and hence need to be regularized. Prior knowledge is usually included via constraints on the network, physics-informed loss functions, or feature preprocessing[40, 55]. Ref. 18 treats the procedure of solving the KS equations as a differentiable program and trains an XC functional using a loss function of density and energy. By backpropagating, the KS equations work as an implicit regularizer for the model. It learns to sample and generate a trajectory from the initial guess density to the exact density during the self-consistent cycle. This improves generalization compared to direct ML models without the KS scheme, such as the KRR models described above, as these models use only the final step results for training and have little information about initial densities.

The success of the KSR model is apparent from the high accuracy achieved for stretched systems. In Fig. 11, the entire dissociation curve of the H_2 molecule is reproduced with chemical accuracy by training at just two separations. A similar performance was reported for H_4. Inclusion of

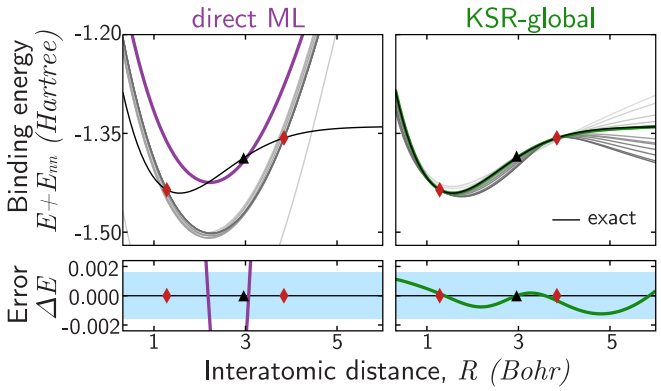


FIG. 11. One-dimensional H_2 dissociation curve, similar to Fig. 5 but with with DMRG data instead of KS-DFT. The colored curves are the optimal models trained on two configurations (red diamonds) and validated on $R = 3$ (black triangle). An ML model directly predicting E from geometries quickly overfits the training data. But the global KSR functional improves with each iteration of the KS equations (grey lines). The lower panel shows the KSR predictions are within the chemical accuracy limit (light blue region). Adapted from Fig. 1 of Ref. 18. Copyright 2021 American Physical Society, licensed under Creative Commons Attribution 4.0 International.

the density loss term generates a much better prediction for the density and the XC potential compared to energy loss alone. The KSR is transferrable in the sense that it could also predict energies for H_2^+ or two H_2 molecules, even though the model was never exposed to those molecules. A successful extrapolation of this method for 3D real molecules may hold the key for a generalizable practical ML density functional which can surpass the accuracy of any human-designed functional.

5. OUTLOOK

In the arena of OF-DFT, a natural question has arisen. If we can find sufficiently accurate force fields by training on DFT (or better) data, why do we need orbital-free DFT? Won't a force field always be much faster (even if slower than simpler force fields)? The current answer is: maybe. For some specific but very important limited cases, ML force fields are both faster and do not run into difficulties. However, there are problematic configurations that current force fields cannot resolve[56]. Moreover, a DFT calculation can be performed for any combination of any atoms in any configuration, whereas most force fields are designed either for exploring materials configuration space with one or two elements, or chemical compound space with about a dozen different elements relevant to medicinal chemistry. A few DFT runs on new combinations of elements and configurations would be cheaper than the cost of new training. Between these two extremes, there is likely room for orbital-free ML-DFT.

But the main focus is to improve XC approximations. Here, there are two distinct areas. For the weakly correlated systems most often encountered in chemistry and many materials, substantial improvements in accuracy would be incredibly useful and might be achievable by finding better combinations of the many approximate functionals already suggested. For strongly correlated systems (including complete dissociation curves of molecules), going beyond the usual semi-local starting points is likely a requirement, and here, the advantage of ML to create entirely non-local functionals is clear.

Possibly the greatest challenge to creating fully non-local functionals is that of generalizability. We need approximations that can be applied to systems of effectively arbitrary size and boundary conditions (open or periodic). A functional that uses the entire density throughout the system is so sophisticated that training on densities of one molecule is unlikely to yield great accuracy on another, and so must be retrained for every case. Yet the very simplest and oldest XC approximation, local exchange[57], generalizes perfectly, by virtue of using only the density at each point to determine its contribution to the XC energy. An ML functional that uses the density within a given radius of the point might improve accuracies for weakly correlated systems, but is unlikely to avoid catastrophic failures for strong correlation. The search for the elusive XC functional will continue, but now includes machine learning alternatives to human designs.

AUTHORS' INFORMATION

Corresponding Author

Kieron Burke, University of California, Irvine;

https://orcid.org/0000-0002-6159-0054;

E-mail: kieron@uci.edu;

Website: dft.uci.edu.

Notes

The authors declare no competing financial interests.

Biographies

Bhupalee Kalita holds a B.S. from Gauhati University and an M.S. from the University of Hyderabad in chemistry. She is currently pursuing a Ph.D. in theoretical chemistry at the University of California, Irvine.

Li Li holds a B.S. in physics from Fudan University, and a Ph.D. in physics from the University of California, Irvine. He works on machine learning and its application in physical sciences at the Google Accelerated Science team.

Ryan J. McCarty holds an M.S. in chemical engineering and a Ph.D. in geological and environmental sciences from Stanford University. He is a University of California President's Postdoctoral Fellow at the University of California, Irvine.

Kieron Burke is a Chancellor's Professor in Chemistry and Physics at the University of California, Irvine, and has wasted much of his life trying to understand and improve density functional theory.

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under Grant No. DGE 1633631 (B.K.) and CHE 1856165 (R.J.M, K.B.).

[1] K. Burke, The Journal of Chemical Physics **136**, 150901 (2012), https://doi.org/10.1063/1.4704546.

[2] J. P. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996).

[3] L. Goerigk, A. Hansen, C. Bauer, S. Ehrlich, A. Najibi, and S. Grimme, Phys. Chem. Chem. Phys. **19**, 32184 (2017).

[4] K. He, X. Zhang, S. Ren, and J. Sun, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

[5] D.-O. Won, K.-R. Müller, and S.-W. Lee, Science Robotics **5**, 10.1126/scirobotics.abb9764 (2020), https://robotics.sciencemag.org/content/5/46/eabb9764.full.pdf.

[6] E. Moen, D. Bannon, T. Kudo, W. Graf, M. Covert, and D. Van Valen, Nature Methods **16**, 1233 (2019).

[7] C. Huntingford, E. S. Jeffers, M. B. Bonsall, H. M. Christensen, T. Lees, and H. Yang, Environmental Research Letters **14**, 10.1088/1748-9326/ab4e55 (2019).

[8] O. T. Unke, S. Chmiela, H. E. Sauceda, M. Gastegger, I. Poltavsky, K. T. Schütt, A. Tkatchenko, and K.-R. Müller, Machine learning force fields (2020), arXiv:2010.07067 [physics.chem-ph].

[9] R. Jinnouchi, J. Lahnsteiner, F. Karsai, G. Kresse, and M. Bokdam, Phys. Rev. Lett. **122**, 225701 (2019).

[10] R. M. Dreizler and E. K. U. Gross, *Density Functional Theory: An Approach to the Quantum Many-Body Problem* (Springer Berlin Heidelberg, Berlin, Heidelberg, 1990).

[11] J. C. Snyder, M. Rupp, K. Hansen, K.-R. Müller, and K. Burke, Phys. Rev. Lett. **108**, 253002 (2012).

[12] J. C. Snyder, M. Rupp, K. Hansen, L. Blooston, K.-R. Müller, and K. Burke, The Journal of Chemical Physics **139**, 224104 (2013), https://doi.org/10.1063/1.4834075.

[13] L. Li, J. C. Snyder, I. M. Pelaschier, J. Huang, U.-N. Niranjan, P. Duncan, M. Rupp, K.-R. Müller, and K. Burke, International Journal of Quantum Chemistry **116**, 819 (2016), https://onlinelibrary.wiley.com/doi/pdf/10.1002/qua.25040.

[14] F. Brockherde, L. Vogt, L. Li, M. E. Tuckerman, K. Burke, and K.-R. Müller, Nature Communications **8**, 872 (2017).

[15] K. Burke and L. O. Wagner, International Journal of Quantum Chemistry **113**, 96 (2013), https://onlinelibrary.wiley.com/doi/pdf/10.1002/qua.24259.

[16] R. Nagai, R. Akashi, and O. Sugino, npj Computational Materials **6**, 43 (2020).

[17] S. Dick and M. Fernandez-Serra, Nature Communications **11**, 3509 (2020).

[18] L. Li, S. Hoyer, R. Pederson, R. Sun, E. D. Cubuk, P. Riley, and K. Burke, Phys. Rev. Lett. **126**, 036401 (2021).

[19] M.-C. Kim, E. Sim, and K. Burke, The Journal of Chemical Physics **140**, 18A528 (2014), https://doi.org/10.1063/1.4869189.

[20] L. Li, T. E. Baker, S. R. White, and K. Burke, Phys. Rev. B **94**, 245129 (2016).

[21] A. J. Cohen, P. Mori-Sánchez, and W. Yang, Science **321**, 792 (2008), https://science.sciencemag.org/content/321/5890/792.full.pdf.

[22] A. D. Becke, The Journal of Chemical Physics **98**, 5648 (1993), https://doi.org/10.1063/1.464913.

[23] S. R. White, Phys. Rev. Lett. **69**, 2863 (1992).

[24] K. A. Hallberg, Advances in Physics **55**, 477 (2006), https://doi.org/10.1080/00018730600766432.

[25] S. Wouters and D. Van Neck, The European Physical Journal D **68**, 272 (2014).

[26] E. M. Stoudenmire, L. O. Wagner, S. R. White, and K. Burke, Phys. Rev. Lett. **109**, 056402 (2012).

[27] M. Rupp, O. A. von Lilienfeld, and K. Burke, The Journal of Chemical Physics **148**, 241401 (2018), https://doi.org/10.1063/1.5043213.

[28] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer New York, New York, NY, 2009) pp. 191–218.

[29] K. Hornik, Neural Networks **4**, 251 (1991).

[30] B. K. Spears, J. Brase, P.-T. Bremer, B. Chen, J. Field, J. Gaffney, M. Kruse, S. Langer, K. Lewis, R. Nora, J. L. Peterson, J. Jayaraman Thiagarajan, B. Van Essen, and K. Humbird, Physics of Plasmas **25**, 080901 (2018), https://doi.org/10.1063/1.5020791.

[31] D. J. Tozer, V. E. Ingamells, and N. C. Handy, The Journal of Chemical Physics **105**, 9200 (1996), https://doi.org/10.1063/1.472753.

[32] S. Manzhos, Machine Learning: Science and Technology **1**, 013002 (2020).

[33] K. Vu, J. C. Snyder, L. Li, M. Rupp, B. F. Chen, T. Khelif, K.-R. Müller, and K. Burke, International Journal of Quantum Chemistry **115**, 1115 (2015).

[34] B. Schölkopf, A. Smola, and K.-R. Müller, in *Artificial Neural Networks — ICANN'97*, edited by W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud (Springer Berlin Heidelberg, Berlin, Heidelberg, 1997) pp. 583–588.

[35] J. C. Snyder, M. Rupp, K.-R. Müller, and K. Burke, International Journal of Quantum Chemistry **115**, 1102 (2015), https://onlinelibrary.wiley.com/doi/pdf/10.1002/qua.24937.

[36] K. Yao and J. Parkhill, Journal of Chemical Theory and Computation **12**, 1139 (2016), pMID: 26812530, https://doi.org/10.1021/acs.jctc.5b01011.

[37] J. Seino, R. Kageyama, M. Fujinami, Y. Ikabata, and H. Nakai, The Journal of Chemical Physics **148**, 241705 (2018), https://doi.org/10.1063/1.5007230.

[38] P. Golub and S. Manzhos, Phys. Chem. Chem. Phys. **21**, 378 (2019).

[39] J. Sun, A. Ruzsinszky, and J. P. Perdew, Phys. Rev. Lett. **115**, 036402 (2015).

[40] J. Hollingsworth, L. Li, T. E. Baker, and K. Burke, The Journal of Chemical Physics **148**, 241743 (2018), https://doi.org/10.1063/1.5025668.

[41] T. E. Markland and M. Ceriotti, Nature Reviews Chemistry **2**, 109 (2018).

[42] K. Raghavachari, G. W. Trucks, J. A. Pople, and M. Head-Gordon, Chemical Physics Letters **157**, 479 (1989).
[43] M. Bogojeski, L. Vogt-Maranto, M. E. Tuckerman, K.-R. Müller, and K. Burke, Nature Communications **11**, 5223 (2020).
[44] S. Dick and M. Fernandez-Serra, The Journal of Chemical Physics **151**, 144102 (2019), https://doi.org/10.1063/1.5114618.
[45] F. L. Hirshfeld, Theoretica chimica acta **44**, 129 (1977).
[46] C. A. Custódio, É. R. Filletti, and V. V. Franca, Scientific Reports **9**, 1886 (2019).
[47] J. Nelson, R. Tiwari, and S. Sanvito, Phys. Rev. B **99**, 075132 (2019).
[48] J. Schmidt, C. L. Benavides-Riveros, and M. A. L. Marques, The Journal of Physical Chemistry Letters **10**, 6425 (2019).
[49] X. Lei and A. J. Medford, Phys. Rev. Materials **3**, 063801 (2019).
[50] Q. Liu, J. Wang, P. Du, L. Hu, X. Zheng, and G. Chen, The Journal of Physical Chemistry A **121**, 7273 (2017).
[51] M. Fritz, M. Fernández-Serra, and J. M. Soler, The Journal of Chemical Physics **144**, 224101 (2016), https://doi.org/10.1063/1.4953081.
[52] K. Ryczko, D. A. Strubbe, and I. Tamblyn, Phys. Rev. A **100**, 022512 (2019).
[53] A. Ryabov, I. Akhatov, and P. Zhilyaev, Scientific Reports **10**, 8000 (2020).
[54] H. Ji and Y. Jung, The Journal of Chemical Physics **148**, 241742 (2018), https://doi.org/10.1063/1.5022839.
[55] K. T. Schütt, M. Gastegger, A. Tkatchenko, K.-R. Müller, and R. J. Maurer, Nature Communications **10**, 5024 (2019).
[56] S. N. Pozdnyakov, M. J. Willatt, A. P. Bartók, C. Ortner, G. Csányi, and M. Ceriotti, Physical Review Letters **125**, 10.1103/physrevlett.125.166001 (2020).
[57] P. A. M. Dirac, Mathematical Proceedings of the Cambridge Philosophical Society **26**, 376–385 (1930).