By-passing the Kohn-Sham equations with machine learning

Felix Brockherde,[1,2] Li Li,[3] Kieron Burke,[4,3,*] and Klaus-Robert Müller[1,5,*]

[1] *Machine Learning Group, Technische Universität Berlin, Marchstr. 23, 10587 Berlin, Germany*
[2] *Max-Planck-Institut für Mikrostrukturphysik, Weinberg 2, 06120 Halle, Germany*
[3] *Departments of Physics and Astronomy, University of California, Irvine, CA 92697, USA*
[4] *Departments of Chemistry, University of California, Irvine, CA 92697, USA*
[5] *Department of Brain and Cognitive Engineering, Korea University, Anam-dong, Seongbuk-gu, Seoul 136-713, Republic of Korea*

(Dated: September 9, 2016)

Last year, at least 30,000 scientific papers used the Kohn-Sham scheme of density functional theory to solve electronic structure problems in a wide variety of scientific fields, ranging from materials science to biochemistry to astrophysics. Machine learning holds the promise of learning the kinetic energy functional via examples, by-passing the need to solve these equations. This should yield substantial savings in computer time, allowing either larger systems or longer time-scales to be tackled. Attempts to machine-learn this functional have been limited by the need to find its derivative. The present work overcomes this difficulty, by learning the density-potential map directly. Both the improved accuracy and lower computational cost is demonstrated on DFT calculations of small molecules.

INTRODUCTION

Kohn-Sham density functional theory[1] is now enormously popular as an electronic structure method in a wide variety of fields[2]. Useful accuracy is achieved with standard exchange-correlation approximations, such as generalized gradient approximations[3] and hybrids[4]. Such calculations are playing a key role in the materials genome initiative[5], at least for weakly correlated materials[6].

There has also been a recent spike of interest in applying machine learning (ML) methods in the physical sciences[7–11]. The majority of these applications involve predicting properties of molecules or materials from large databases of KS-DFT calculations[12–14]. A few applications involve finding potential energy surfaces within MD simulations[15]. Fewer still have focussed on finding the functionals of DFT as a method of performing KS electronic structure calculations without solving the KS equations[16–18]. If such attempts could be made practical, the possible speed-up in repeated DFT calculations of similar species, such as occur in ab initio MD simulations in water, is enormous.

A key difficulty has been the need to extract the functional derivative of the non-interacting kinetic energy. The non-interacting kinetic energy functional $T_s[n]$ of the density n is used in *two* distinct ways in a KS calculation[1], as illustrated in Fig. 1: (i) its functional derivative is used in the Euler equation which is solved in the self-consistent cycle and (ii) when self-consistency is reached, the ground-state energy of the system is calculated by $E[n]$, an Orbital-Free (OF) mapping. The solution of the KS equations performs both tasks exactly. Early results on simple model systems showed that machine learning could provide highly accurate values for $T_s[n]$ with only modest amounts of training[16], but that the corresponding functional derivatives are too noisy to yield sufficiently accurate results to (i). Subsequent schemes overcome this difficulty in various ways, but typically lose a factor of 10 or more in accuracy[17], and their computational cost can increase dramatically with system complexity.

Here we present an alternative ML approach, in which we replaced the Euler equation by directly learning the Hohenberg-Kohn (HK) map $v(\mathbf{r}) \to n(\mathbf{r})$ (red dashed in Fig. 1) from the one-body potential of the system of interest to the interacting ground-state density, i.e. we establish an ML-HK map. We show that this map can be learned at a much more modest cost than either previous ML approaches to find the functional and its derivative (ML-OF) or direct attempts to model the energy as a functional of $v(\mathbf{r})$ (ML-KS). Furthermore we show that it can immediately be applied to molecular calculations, by calculating the properties of both H_2 and H_2O. Moreover, since we have already implemented this with a standard quantum chemical code (Quantum Espresso[19]) using a standard DFT approximation (PBE), this can now be tried on much larger scales.

RESULTS

We will first outline theoretical results, most prominently the ML-HK map, and then do simulations of 1-D systems and 3-D molecules.

ML-Hohenberg-Kohn map

Previous results show that for an ML-OF approach, the accuracy of ML kinetic energy models $T_s^{\mathrm{ML}}[n]$ improve rapidly with the amount of data. But minimizing the total energy via gradient descent requires the calculation

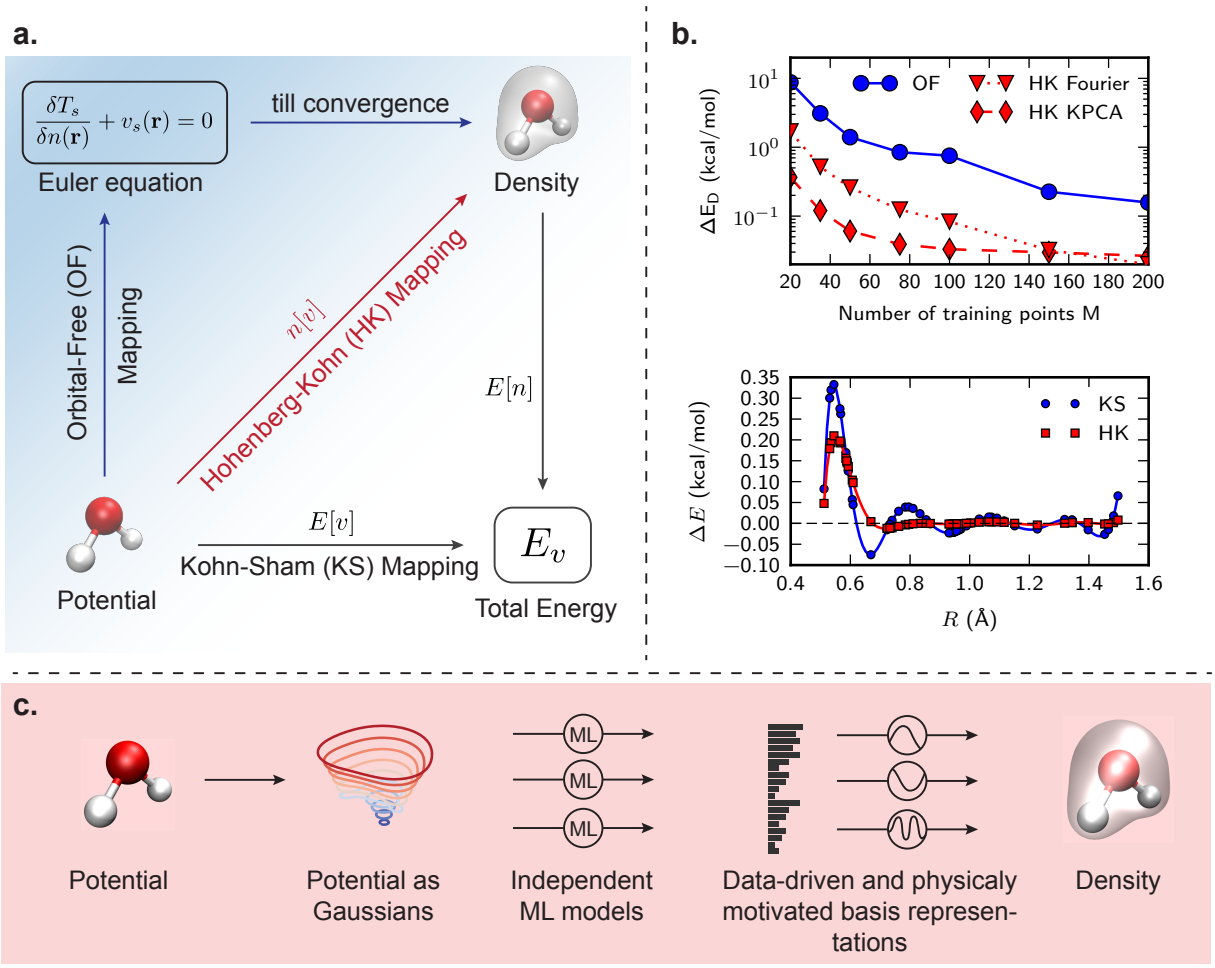

Figure 1. **a.** Mappings used in this paper. The bottom arrow represents $E[v]$, a conventional electronic structure calculation, i.e., KS-DFT. The ground state energy is found by solving KS equations given the external potential, v. $E[n]$ is the total energy density functional. The red arrow is the HK map $n[v]$ from external potential to its ground state density. **b** top. How ΔE_D depends on M for ML-OF and ML-HK with different basis sets for the 1-D problem. **b** bottom. Errors of the ML-KS map and the ML-HK map for different numbers of training data points M (in kcal/mol) for H_2. **c**. How our Machine Learning Hohenberg-Kohn (ML-HK) map makes predictions. The molecular geometry is represented by Gaussians; many independent Kernel Ridge Regression models predict each basis coefficient of the density. We analyze the performance of data-driven (ML) and common physical basis representations for the electron density.

of the *gradient* of the kinetic energy model T_s^{ML} (see Fig. 1). Calculating this gradient is challenging. Due to the data driven nature of, e.g., kernel models, the machine-learned kinetic energy functional has no information in directions that point outside the data manifold[20]. This heavily influences the gradient to an extent that it becomes unusable without further processing[16]. There have been several suggestions to remedy this problem but all of them share a significant loss in accuracy compared to $T_s[n]$[17, 18, 21].

However, we propose an interesting alternative to gradients and the ML-OF approach. Recently, it has been shown that the Hohenberg-Kohn map for the density as a functional of the potential can be approximated extremely accurately using semiclassical expressions[22]. Such expressions do not require the solution of any differential equation, and become more accurate as the number of particles increases. Errors can be negligible even for just 2 distinct occupied orbitals.

Inspired by this success, we suggest to circumvent the kinetic energy gradient and directly train a multivariate machine learning model. We name this the ML-Hohenberg-Kohn (ML-HK) map:

$$n^{\mathrm{ML}}[v](x) = \sum_{i=1}^{M} \beta_i(x) k(v, v_i). \tag{1}$$

Here, each density grid point is associated with a group of model weights $\boldsymbol{\beta}$. Training requires solving an optimization problem for each density grid point. While this

is possible in 1-D, it rapidly becomes intractable in 3-D, since the number of grid points grows cubically.

The use of a basis representation for the densities, as in

$$n^{\mathrm{ML}}[v](x) = \sum_{l=1}^{L} u^{(l)}[v]\phi_l(x), \tag{2}$$

renders the problem tractable even for 3-D. A machine learning model that predicts the basis function coefficients $u^{(l)}[v]$ instead of the grid points is then formulated.

Predicting the basis function coefficients not only makes the machine learning model efficient and allows the extension of the approach to 3-D but also permits regularization, e.g. to smoothen the predicted densities by removing the high frequency basis functions for example, or to further regularize the machine learning model complexity for specific basis functions.

For orthogonal basis functions, the machine learning model reduces to several independent regression models and admits an analytical solution analogous to Kernel Ridge Regression (see supplement Eq. 4):

$$\boldsymbol{\beta}^{(l)} = (K_{\sigma^{(l)}} + \lambda^{(l)} I)^{-1}\mathbf{u}^{(l)}, \quad l = 1, \ldots, L. \tag{3}$$

Here, for each basis function coefficient, $\mathbf{u}^{(l)}$ and $\lambda^{(l)}$ are regularization parameters and $K_{\sigma^{(l)}}$ is a Gaussian kernel with kernel width $\sigma^{(l)}$. The $\lambda^{(l)}$ and $\sigma^{(l)}$ can be chosen individually for each basis function via independent cross-validation (see [12, 23]). This ML for the HK model avoids prior gradient descent procedures and with it the necessity to "de-noise" the gradients. Due to the independence of Eq. 3 for each l, the solution scales nicely.

Functional and Density driven error

How can the performance of the ML-HK map be measured? It has recently been shown how to separate out the effect of the error in the many-body effects functional F and the error in the density $n(\mathbf{r})$ on the resulting error in the total energy of any approximate, self-consistent DFT calculation[24]. Let $\tilde{F}$ be the approximation of the many body functional F, and $\tilde{n}(\mathbf{r})$ the approximate ground state density when the approximate many body functional is used in the Euler equation. Defining $\tilde{E}[n] = \tilde{F}[n] + \int d^3r n(\mathbf{r})v(\mathbf{r})$ yields

$$\Delta E = \tilde{E}[\tilde{n}] - E[n] = \Delta E_F + \Delta E_D \tag{4}$$

where $\Delta E_F = \tilde{E}[n] - E[n]$ is the functional-driven error, while $\Delta E_D = \tilde{E}[\tilde{n}] - \tilde{E}[n]$ is the density-driven error. In most calculations, the error is dominated by the functional-driven error. The standard approximations can, in some specific DFT calculations, produce abnormally large density errors that dominate the total error. In such situations, using a more accurate density can greatly improve the result [24–26]. We will use these definitions to measure the accuracy of the ML-HK map.

1-D potentials

The following results demonstrate how much more accurate ML is when applied directly to the HK map. The box problem originally introduced in Snyder *et al.* [16] is used to illustrate the principle. Random potentials consisting of three Gaussian dips were generated inside a hard-wall box of length 1 (atomic units), and the Schrödinger equation for non-interacting electrons solved extremely precisely. Up to 200 cases were used to train an ML model for $T_{\mathrm{s}}[n]$.

To measure the accuracy of an approximate HK map, the analysis of the previous section is applied to the KS DFT problem. For truly non-interacting electrons, the many-body functional F is just the kinetic energy, and the functional error is simply

$$\Delta E_F = \tilde{T}_{\mathrm{s}}[n] - T_{\mathrm{s}}[n], \tag{5}$$

i.e., the error made in an approximate functional on the exact density. Table I on the left gives the errors made by ML-OF for the total energy, and its different components. The density-driven contribution to the error is always comparable to, or greater than, the functional-driven error. This means the calculation is abnormal, and can be significantly improved by using a more accurate density. As M grows, the error becomes completely dominated by the error in the density. This shows that the largest source of error is in using the ML approximation to T_{s} to solve the Euler equation to find the density.

Now this separation of errors can be used to analyze the direct ML-HK map. The most accurate $E^{\mathrm{ML}}[n]$ model, in this case the one trained on 200 data points, is taken and the density-driven error of the ML-HK map is calculated. The average error of $E^{\mathrm{ML}}[n]$ on the exact density is only 0.04 kcal/mol, as given by ΔE_F. For $M \leq 50$, the error in the $E^{\mathrm{ML}}[n]$ map is negligible relative to the density-driven error, so the total energy error is almost identical to the density-driven error here.

Three variants of the ML-HK map were tested. First, direct prediction of the grid coefficients: In this case, $\mathbf{u}_i^{(l)} = n_i(x_l)$, $l = 1, \ldots, G$. This variant is tested in 1-d only; in 3-D the high dimensionality will be prohibitive. 500 grid points were used, as in Snyder *et al.* [16]. Second, a common Fourier basis is tested. The density can be transformed efficiently via the discrete Fourier transform, using 49 Fourier basis functions in total. In 3-D these basis functions correspond to plane waves. The

	ML-OF $E[v]$						ML-HK $n[v]$							
	ΔE		ΔE_F		ΔE_D		ΔE grid		ΔE_D grid		ΔE_D Fourier		ΔE_D KPCA	
M	MAE	max	MAE	max	MAE	max	MAE	max	MAE	max	MAE	max	MAE	max
20	7.7	47	7.7	60	8.8	87	2.15	11.5	2.14	11.5	1.7	9.8	0.37	5.5
50	1.6	30	1.3	7.3	1.4	31	0.26	2.4	0.26	2.4	0.26	2.4	0.061	0.96
100	0.74	17	0.2	2.6	0.75	17	0.084	0.82	0.081	0.82	0.083	0.81	0.033	0.43
200	0.16	2.9	0.039	0.6	0.16	2.9	0.042	0.059	0.019	0.45	0.02	0.45	0.026	0.26

Table I. Errors (kcal/mol) on the 1-D dataset for different numbers of training data, M. The ML-HK errors are calculated using the ML-OF functional with 200 points.

back-projection $\mathbf{u} \mapsto n$ to input space is simple, but although the basis functions are physically motivated, they are very general and not specifically tailored to density functions. This is a motivation for exploring, third, a Kernel PCA (KPCA) basis[27]. KPCA[28] is a popular generalization of PCA that yields basis functions that maximize variance in a higher dimensional feature space. The KPCA basis functions are data-driven and computing them requires an eigen-decomposition of the Kernel matrix. Good results are achieved with only 25 KPCA basis functions. The KPCA approach gives better results because it can take the non-linear structure in the density space into account. However, it introduces the pre-image problem: It is not trivial to project the densities from KPCA space back to their original (grid) space (see supplement). It is thus not immediately applicable to 3-D applications.

The results are shown on the right-hand side of Table I. For all values of M, and all three variants, the average density-driven errors are typically an order of magnitude smaller than their counterparts on the left (ML-OF). This demonstrates that direct learning of the HK map is much easier than deducing that map from a machine-learned kinetic-energy functional. The table also shows the corresponding errors in energy for the ML-HK map, when combined with the highly-accurate $M = 200$ energy map. These, again, are much smaller than their counterparts for the ML-OF map, for the same reasons.

The number of Fourier basis functions can be reduced to 30 without getting significantly higher errors. The mean average error ΔE_D for $M = 100$ increases from 0.083 to 0.93. Further reduction leads to inaccuracies in the T_s^ML predictions.

3-D molecules

We next apply the ML-HK approach to realistic DFT calculations of small molecules. All the methods, except the new one, applied in the previous section to the box problem become prohibitively expensive in 3-D, because the gradient descent procedure does not converge even with unrealistically many training points. Thus we can not compare to the ML-OF approach anymore and instead compare the ML-HK map to a ML-KS approach. We again measure the density-driven error of the ML-HK map:

$$\overline{|\Delta E_D|} = \frac{1}{K}\sum_{k=1}^{K}|E^\mathrm{ML}[n_k] - E^\mathrm{ML}[n^\mathrm{ML}[v_k]]|. \qquad (6)$$

Since the total energy functional is unknown, but necessary to compute the density-driven error, an ML model $E^\mathrm{ML}[n]$ is trained for it. This map $E^\mathrm{ML}[n]$ is trained on significantly more training points than $n^\mathrm{ML}[n]$ to be as accurate as possible and yields errors far smaller than any of the maps being tested.

We can not define a density-driven error of the ML-KS map and therefore directly measure the total energy error

$$\overline{|\Delta E|} = \frac{1}{K}\sum_{k=1}^{K}|E^\mathrm{ML}[v_k] - E_{v_k}|, \qquad (7)$$

where E_{v_k} is the result of the KS calculation for the external potential $v_k(r)$.

Both approaches require the characterization of the Hamiltonian by its external potential. The external (Coulomb) potential diverges for the 3-D molecules and is therefore not a good feature to measure the distance in ML. Instead, we use an artificial *Gaussians potential* in the form of

$$v(r) = \sum_{\alpha=1}^{N^a} Z_\alpha \exp\left(\frac{-\|r - R_i\|^2}{2\gamma^2}\right) \qquad (8)$$

where R_i are the positions and Z_α are the nuclear charges of the N^a atoms. The Gaussian potential is used for the ML representation only. The width γ is a hyperparameter of the algorithm. The choice is arbitrary but can be cross-validated. We find good results with $\gamma = 0.2$Å. The idea of using Gaussians to represent the external potential has been used previously[29]. The Gaussians potential is discretized on a course grid with grid spacing $\Delta = 0.08$Å.

Our first prototype is H_2 as the simplest molecule. Let R be the distance between the H atoms. A dataset of 150 geometries is created by varying R between 0.5 and 1.5 angstrom (sampled uniformly). This dataset is split into a *grand training set* of 100 geometries and a test set of 50 geometries. The test set is unseen by the ML algorithms and only used to measure the *out-of-sample* test error. We report only out-of-sample errors which are always measured on the complete test set.

The $E^{\mathrm{ML}}[n]$ used to measure the accuracy of the ML-HK map is trained on the full grand training set. To evaluate the performance of the ML-KS map and the ML-HK map, subsets of varying sizes M are chosen out of the grand training set to train the $E^{\mathrm{ML}}[v]$ and $n^{\mathrm{ML}}[v]$ models on. Because the required training subsets are so small, careful selection of a subset that covers the complete range of R is necessary. It is ensured by selecting the M training points out of the grand training set so that the R values are nearly equally spaced (see supplement for details).

	ML-KS $E^{\mathrm{ML}}[v]$				ML-HK $E^{\mathrm{ML}}[n^{\mathrm{ML}}[v]]$			
	ΔE		ΔR_o	$\Delta\theta_0$	ΔE		ΔR_o	$\Delta\theta_0$
M	MAE	max			MAE	max		
				H_2				
5	1.7	5.5	2.6	-	0.25	0.99	0.17	-
7	0.37	1.4	0.23	-	0.038	0.21	0.070	-
10	0.060	0.33	0.18	-	0.024	0.14	0.071	-
				H_2O				
15	0.12	0.51	0.16	0.20	0.088	0.44	0.10	0.31
20	0.10	0.47	0.13	0.21	0.013	0.086	0.020	0.14

Table II. Prediction errors on 3-D molecules versus M, the number of training points, on a hold-out set for the ML-HK map ($E^{\mathrm{ML}}[v]$ via Eq. 7) and the ML-KS map ($E^{\mathrm{ML}}[n^{\mathrm{ML}}[v]]$ via Eq. 6). Energies in kcal/mol, bond-lengths in pm, and angles in degrees.

The performance of the ML-HK map and ML-KS map is compared by training $E^{\mathrm{ML}}[v]$ that maps from Gaussians potential to total energy and $n^{\mathrm{ML}}[v]$ that maps from Gaussians potential to the ground-state density in Fourier basis representation ($l = 25$). The models are evaluated as in Eq. 7 and 6. The prediction errors for H_2 are listed in Table II. With $M = 7$ training data, the MAE of the ML-HK map is one order of magnitude smaller than of the ML-KS map. This indicates that even in 3-D, the potential-density relationship via the HK map is much easier to learn than the potential-energy relationship via the KS map.

Fig. 1 shows the errors made by the ML-OF and the ML-HK maps. The equilibrium bond-distance is $R_0 = 0.74$Å. The error of the HK is much smaller and smoother, except when the separation becomes very small. The machine learning models are trained on reference data from KS-DFT calculations with PBE exchange-correlation. The mean average error that is introduced by the PBE approximation on our H_2 dataset is 2.3 kcal/mol (compared to exact calculations), i.e., it is well above the errors of the ML model and verifies that the error introduced by the ML-HK map is negligible in a DFT calculation.

The next prototype is H_2O, a more complicated 3-D molecule that is parametrized with 3 degrees of freedom: two bond lengths and a bond angle. To create a dataset, the equilibrium configuration ($R_0 = 0.97$Å$, \theta_0 = 104.2^\circ$ using PBE) is taken and each bond length is varied by a uniformly sampled value between ± 0.075 angstrom away from R_0. The angle θ is also varied between ± 8.59 degrees (± 0.15 rad) away from θ_0. In total 350 geometries are sampled this way (see supplement for a visualization of the sampling range). A random subset of 50 geometries is taken as the out-of-sample test set and the remaining 300 geometries are taken as the grand training set just as for H_2. Again, $E^{\mathrm{ML}}[v]$ is trained on the full grand training set. Because there are now 3 parameters, it is harder to select equidistant samples for the training subset of M data points. We therefore use K-means to find M clusters and select for each cluster the geometry of the grand training set closest to that cluster's center for the training subset (see supplement for details).

Models are trained just as for H_2 and the results are given in Table II. As expected, due to the higher degree of freedom in H_2O compared to H_2, a larger training set size M is required. However, even for the more complicated molecule the ML-HK map works consistently better than the ML-KS map, and provides an improved potential energy surface, as shown in Fig. 2.

The PBE mean average error on the H_2O dataset is 1.2 kcal/mol. Again, it can be concluded that ML does not introduce a new significant source of error.

Geometry Optimization

The ML-HK map can also be used to find the minimum energy geometry configuration. The total energy is minimized as the geometry varies with respect to both bond lengths and angles. For optimization, we use Powell's method [30], which requires a starting point and an evaluation function to be minimized. A random geometry from the training set is used as starting point. The evaluation functions are $E^{\mathrm{ML}}[v]$ and $E^{\mathrm{ML}}[n^{\mathrm{ML}}[v]]$. For the H_2O case, the search is restricted to symmetric configurations. This optimization consistently converges to the correct minima regardless of starting point, consistent with the maps being convex, i.e., the potential energy curves are sufficiently smooth as to avoid accidental local minima.

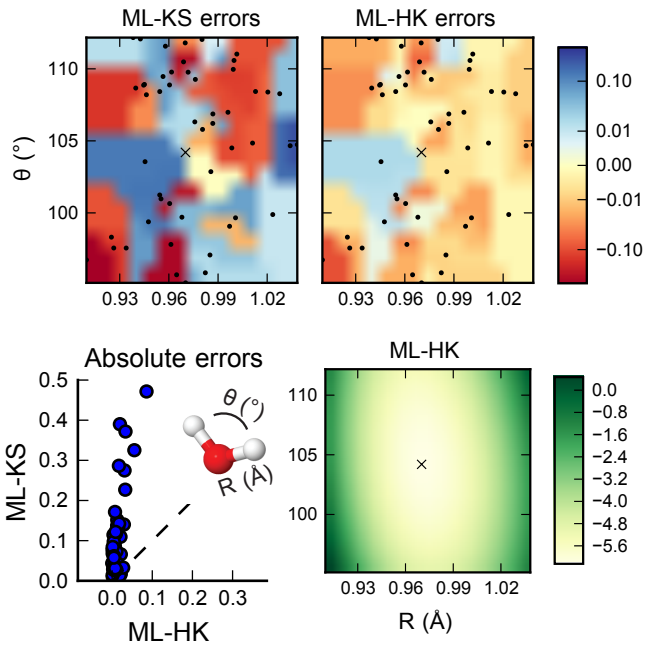

Figure 2. **Top.** Distribution of errors against PBE on the H_2O dataset for ML-KS $E^{\mathrm{ML}}[v]$ and ML-HK $E^{\mathrm{ML}}[n^{\mathrm{ML}}[v]]$. The errors are plotted on a symmetric log scale with linear threshold of 0.01, using nearest neighbor interpolation for coloring. Black dots mark the test set geometries. **Bottom left.** Comparison of the PBE errors made by ML-HK and ML-KS on the test set geometries. **Bottom right.** Energy landscape of the ML-HK map (R against θ, colorbar is offset by -1.3812×10^4 kcal/mol). All models trained on $M = 20$ training points. Energies and errors in kcal/mol. A black cross marks the PBE equilibrium position.

DISCUSSION

For several decades, density functional theory has been a cross-disciplinary area between theoretical physics, chemistry, and materials sciences. The methods used, and advances made, in each field have cross-fertilized comparable and even better advances in the other fields. This has led to its enormous popularity and widespread success, despite its well-known limitations in both accuracy and the systems and properties to which it can be applied.

The present work makes a key step forward toward adding an entirely new ingredient to this mix, namely the construction of functionals via machine learning. While previous work showed proofs of principle in 1D, this is the first demonstration in 3D, using real molecules and production-level codes. This opens the possibility that machine-learning methods, which complement all existing approaches to functional approximation, could become a new and very different approach to this problem, with the potential to greatly reduce the computational cost of routine DFT calculations.

Our new method, directly learning the Hohenberg-Kohn density-potential map, overcomes a key bottleneck in previous methodologies, that only became apparent when going beyond 1D calculations. This approach avoids solving an intermediate more general problem (the gradient descent) to find the solution of the more specific problem (finding the ground-state density). This is called transductive inference by the machine learning community and is thought to be key to successful statistical inference methods[31].

We also confirmed that following a direct prediction approach with the ML-HK map increases the accuracy consistently on both 1-D examples and small 3-D molecules. We are also able to learn density models that outperform energy models trained on much more training data. This quantitative observation allows us to conclude that learning density models is much easier than learning energy models. Such a finding should be no surprise to practitioners of the art of functional construction (see, e.g., [22]), but the present work quantifies this observation using standard statistical methods.

We have also derived a way to use basis functions to make the approach computationally feasible. This makes it easier to integrate the method into existing DFT codes. Another advantage is the possibility to take the innate structure of the densities into account, i.e. spatial correlations are preserved by using low frequency basis functions. Again, this fits with the intuition of experienced practioners in this field, but here we have quantified this in terms of machine-learned functionals.

Direct prediction of energies (e.g., the ML-KS map) always has the potential to lead to conceptually easier methods. But such methods must also abandon the insights and effects that have made DFT a practical and usefully accurate tool over the past half century. Many usefully accurate DFT approximations already exist, and the corrections to such approximations can be machine-learned in precisely the same way as the entire functional has been approximated here. If machine-learning corrections requires less data, the method becomes more powerful by taking advantage of existing successes. Furthermore, existing theorems, such as the viral theorem[32], might also be used to directly construct the kinetic energy functional from an ML-HK map. Finally, in the case of orbital-dependent functionals, such as meta-GGA's or global hybrids, the method presented here must be extended to learn, e.g., the full density matrix instead of just the density. All this provides useful directions in which to expand on the results shown here.

METHODS

Kohn-Sham Density Functional Theory (KS-DFT)

Density Functional Theory is a computational electronic structure method that determines the properties of many-body systems by using functionals of the electron density. The foundation is the Hohenberg-Kohn theorem[33] that establishes a one-to-one relationship between potential and density, i.e. at most one potential can give rise to a given ground-state density.

Kohn-Sham DFT avoids direct approximation of many body effects by imagining a fictitious system of non-interacting electrons with the same density as the real one[1]. Its accuracy is limited by the accuracy of existing approximations to the unknown exchange-correlation energy, while its computational bottleneck is the solution of the Kohn-Sham equations that describe the non-interacting particles.

Here, all 3-D DFT calculations are performed with the Quantum ESPRESSO code[34] using the PBE exchange-correlation functional[35] and projector augmented waves (PAWs)[36, 37] with Troullier-Martin pseudization for describing the ionic cores[38]. All relevant parameters are converged to 0.01 kcal/mol.

The 1-D dataset is taken from Snyder *et al.* [16].

Kernel Ridge Regression (KRR)

Kernel ridge regression[39, 40] (KRR) is a machine learning method for regression. It is a kernelized version of Ridge Regression which minimizes the least squares error and applies an ℓ_2 (Tikhonov) regularization. Let $x_1, \ldots, x_m \in \mathbb{R}^d$ be the training data points and let $\mathbf{Y} = (y_1, \ldots, y_m)^T$ be their respective labels. KRR then optimizes

$$\min_{\alpha} \sum_{i=1}^{m} \left| y_i - \sum_{j=1}^{m} \alpha_j k(x_i, x_j) \right|^2 + \lambda \|\boldsymbol{\alpha}\|^2 \qquad (9)$$

where k is the kernel function and λ is a regularization parameter. It admits an analytical solution

$$\boldsymbol{\alpha} = (\mathbf{K} - \lambda \mathbf{I})^{-1} \mathbf{Y}. \qquad (10)$$

$\mathbf{K}$ is the kernel matrix with $\mathbf{K}_{ij} = k(x_i, x_j)$. Most popular is the Gaussian (radial basis function) kernel which allows to find a smooth non-linear model function in input space that corresponds to a linear function in an infinite dimensional feature space[23].

For the ML-Hohenberg-Kohn map, the canonical cost function is given by the $\mathcal{L}_2$ distance between predicted and true densities

$$C(\boldsymbol{\beta}) = \sum_{i=1}^{M} \|n_i - n^{\mathrm{ML}}[v_i]\|_{\mathcal{L}_2} \qquad (11)$$

$$= \sum_{i=1}^{M} \left\| n_i - \sum_{l=1}^{L} \sum_{j=1}^{M} \beta_j^{(l)} k(v_i, v_j) \phi_l \right\|_{\mathcal{L}_2} . \qquad (12)$$

The ML model coefficients $\boldsymbol{\beta}^{(l)}$ can be optimized independently for each basis coefficient l via

$$\boldsymbol{\beta}^{(l)} = (K_{\sigma^{(l)}} + \lambda^{(l)} I)^{-1} \mathbf{u}^{(l)}, \quad l = 1, \ldots, L. \qquad (13)$$

Cross-validation

Note that all model parameters and hyper-parameters are estimated on the training set; the hyper-parameter choice makes use of standard cross-validation procedures (see Hansen *et al.* [12]). Once the model is fixed after training, it is applied unchanged out-of-sample.

Exact calculations

An exact binding energy curve of H_2 is calculated with the Full Configuration Interaction (FCI) method using the Molpro Quantum Chemistry Software[41]. For $\mathsf{H}_2\mathsf{O}$, accurate energies are calculated using CCSD(T)[42].

ACKNOWLEDGMENTS

We thank J.C. Snyder for early discussions and H. Glawe for helpful guidance regarding the 3-D reference computations. We thank IPAM at UCLA for repeated hospitality. Work at UC Irvine supported by NSF CHE-1464795. KRM and FB thank the Einstein Foundation for generously funding the ETERNAL project.

AUTHOR CONTRIBUTIONS

FB performed DFT calculations and ML experiments. LL performed FCI and CCSD(T) calculations. KB and KRM initiated the work and contributed to the theory and experiments. All authors contributed to the manuscript.

* to whom correspondence should be addressed.

[1] W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965).
[2] A. Pribram-Jones, D. A. Gross, and K. Burke, Annual Review of Physical Chemistry **66**, 283 (2015).

[3] J. P. Perdew, Phys. Rev. B **33**, 8822 (1986).
[4] A. D. Becke, The Journal of Chemical Physics **98**, 5648 (1993).
[5] "Materials genome initiative for global competitiveness," (2011).
[6] A. Jain, G. Hautier, C. J. Moore, S. P. Ong, C. C. Fischer, T. Mueller, K. A. Persson, and G. Ceder, Computational Materials Science **50**, 2295 (2011).
[7] Z. D. Pozun, K. Hansen, D. Sheppard, M. Rupp, K.-R. Müller, and G. Henkelman, The Journal of Chemical Physics **136**, 174101 (2012).
[8] R. T. McGibbon and V. S. Pande, Journal of Chemical Theory and Computation **9**, 2900 (2013).
[9] T. L. Fletcher, S. J. Davie, and P. L. Popelier, Journal of chemical theory and computation **10**, 3708 (2014).
[10] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, Phys. Rev. Lett. **108**, 058301 (2012).
[11] G. Hautier, C. C. Fischer, A. Jain, T. Mueller, and G. Ceder, Chem. Mater. **22**, 3762 (2010).
[12] K. Hansen, G. Montavon, F. Biegler, S. Fazli, M. Rupp, M. Scheffler, O. A. von Lilienfeld, A. Tkatchenko, and K.-R. Müller, J. Chem. Theory Comput. **9**, 3404 (2013).
[13] K. T. Schütt, H. Glawe, F. Brockherde, A. Sanna, K.-R. Müller, and E. K. U. Gross, Phys. Rev. B **89**, 205118 (2014).
[14] K. Hansen, F. Biegler, R. Ramakrishnan, W. Pronobis, O. A. von Lilienfeld, K.-R. Müller, and A. Tkatchenko, The Journal of Physical Chemistry Letters **6**, 2326 (2015), pMID: 26113956, http://dx.doi.org/10.1021/acs.jpclett.5b00831.
[15] Z. Li, J. R. Kermode, and A. De Vita, Phys. Rev. Lett. **114**, 096405 (2015).
[16] J. C. Snyder, M. Rupp, K. Hansen, K.-R. Müller, and K. Burke, Phys. Rev. Lett. **108**, 253002 (2012).
[17] 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).
[18] J. C. Snyder, M. Rupp, K. Hansen, L. Blooston, K.-R. Müller, and K. Burke, J. Chem. Phys **139**, 224104 (2013).
[19] P. Giannozzi, S. Baroni, N. Bonini, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, G. L. Chiarotti, M. Cococcioni, I. Dabo, A. Dal Corso, S. de Gironcoli, S. Fabris, G. Fratesi, R. Gebauer, U. Gerstmann, C. Gougoussis, A. Kokalj, M. Lazzeri, L. Martin-Samos, N. Marzari, F. Mauri, R. Mazzarello, S. Paolini, A. Pasquarello, L. Paulatto, C. Sbraccia, S. Scandolo, G. Sclauzero, A. P. Seitsonen, A. Smogunov, P. Umari, and R. M. Wentzcovitch, Journal of Physics: Condensed Matter **21**, 395502 (19pp) (2009).
[20] J. C. Snyder, S. Mika, K. Burke, and K.-R. Müller, in *Empirical Inference*, edited by B. Schölkopf, Z. Luo, and V. Vovk (Springer Berlin Heidelberg, 2013) pp. 245–259.
[21] J. C. Snyder, M. Rupp, K.-R. Müller, and K. Burke, Int. J. Quantum Chem. **115**, 1102 (2015).
[22] R. F. Ribeiro, D. Lee, A. Cangi, P. Elliott, and K. Burke, Phys. Rev. Lett. **114**, 050401 (2015).
[23] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, IEEE Trans. Neural Netw. **12**, 181 (2001).
[24] M.-C. Kim, E. Sim, and K. Burke, Phys. Rev. Lett. **111**, 073003 (2013).
[25] M.-C. Kim, E. Sim, and K. Burke, The Journal of Chemical Physics **140**, 18A528 (2014).
[26] M.-C. Kim, H. Park, S. Son, E. Sim, and K. Burke, J. Phys. Chem. Lett. **6**, 3802 (2015).
[27] B. Schölkopf, S. Mika, C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. Smola, IEEE Trans. Neural Netw. **10**, 1000 (1999).
[28] B. Schölkopf, A. Smola, and K.-R. Müller, Neural Comput. **10**, 1299 (1998).
[29] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, Phys. Rev. Lett. **104**, 136403 (2010).
[30] M. J. D. Powell, The Computer Journal **7**, 155 (1964).
[31] V. Vapnik, *The Nature of Statistical Learning Theory*, Information Science and Statistics (Springer, 2000).
[32] A. Cangi, D. Lee, P. Elliott, K. Burke, and E. K. U. Gross, Phys. Rev. Lett. **106**, 236404 (2011).
[33] P. Hohenberg and W. Kohn, Phys. Rev. **136**, B864 (1964).
[34] P. Giannozzi, S. Baroni, N. Bonini, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, G. L. Chiarotti, M. Cococcioni, I. Dabo, A. D. Corso, S. d. Gironcoli, S. Fabris, G. Fratesi, R. Gebauer, U. Gerstmann, C. Gougoussis, A. Kokalj, M. Lazzeri, L. Martin-Samos, N. Marzari, F. Mauri, R. Mazzarello, S. Paolini, A. Pasquarello, L. Paulatto, C. Sbraccia, S. Scandolo, G. Sclauzero, A. P. Seitsonen, A. Smogunov, P. Umari, and R. M. Wentzcovitch, J. Phys.: Condens. Matter **21**, 395502 (2009).
[35] J. P. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996).
[36] G. Kresse and D. Joubert, Phys. Rev. B **59**, 1758 (1999).
[37] P. E. Blöchl, Phys. Rev. B **50**, 17953 (1994).
[38] N. Troullier and J. L. Martins, Phys. Rev. B **43**, 1993 (1991).
[39] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning - Data Mining, Inference, and Prediction*, 2nd ed., Springer Series in Statistics (Springer, 2009).
[40] 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).
[41] H.-J. Werner, P. J. Knowles, G. Knizia, F. R. Manby, M. Schütz, *et al.*, "Molpro, version 2015.1, a package of ab initio programs," (2015).
[42] T. B. Adler, G. Knizia, and H.-J. Werner, Journal of Chemical Physics **127**, 221106 (2007).