By-passing the Kohn-Sham equations with machine learning

Felix Brockherde,1, 2 Leslie Vogt,3 Li Li,4 Mark E. Tuckerman,3, 5, 6 Kieron Burke,7, 4, 8 and Klaus-Robert Müller1, 8, ∗

1 Machine Learning Group, Technische Universität Berlin, Marchstr. 23, 10587 Berlin, Germany
2 Max-Planck-Institut für Mikrostrukturphysik, Weinberg 2, 06120 Halle, Germany
3 Department of Chemistry, New York University, New York, NY 10003, USA
4 Departments of Physics and Astronomy, University of California, Irvine, CA 92697, USA
5 Courant Institute of Mathematical Science, New York University, New York, NY 10003, USA
6 Department of Chemistry, University of California, Irvine, CA 92697, USA
7 Department of Chemistry, University of California, Irvine, CA 92697, USA
8 Department of Brain and Cognitive Engineering, Korea University, Anam-dong, Seongbuk-gu, Seoul 136–713, Republic of Korea

(Dated: March 3, 2017)

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 the Kohn-Sham equations. This should yield substantial savings in computer time, allowing either larger systems or longer time-scales to be tackled, but attempts to machine-learn this functional have been limited by the need to find its derivative. The present work overcomes this difficulty by directly learning the density-potential and energy-density maps for test systems and various molecules. Both improved accuracy and lower computational cost with this method are demonstrated by reproducing DFT energies for a range of molecular geometries generated during molecular dynamics simulations. Moreover, the methodology could be applied directly to quantum chemical calculations, allowing construction of density functionals of quantum-chemical accuracy.

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[15–17]. Fewer still have focussed on finding the functionals of DFT as a method of performing KS electronic structure calculations without solving the KS equations[18–21]. 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, 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[18], 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[20], 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 $v(r) \rightarrow n(r)$ (red line in Fig. 1a) 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 $v(r)$ (ML-KS). Furthermore we show that it can immediately be applied to molecular calculations, by calculating the energies of small molecules over a range of conformers. Moreover, since we have already implemented this approach with a standard quantum chemical code (Quantum Espresso[22]) using a standard DFT approximation (PBE), this can now be tried on much larger scales.

The ML-HK map reflects the underlying computa-
ntional approach used to generate a particular electron density, but is not restricted to any given electronic structure method. Many molecular properties, not only the energy, are dependent on the electron density, making the ML-HK map more versatile than a direct ML-KS mapping. We also establish that densities can be learned with sufficient accuracy to distinguish between different DFT functionals, providing a route to future functional development by generating precise densities for a range of molecules and conformations.

RESULTS

We will first outline theoretical results, most prominently the ML-HK map, and then illustrate the approach with 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 KS kinetic energy models $T_s^{ML}[n]$ improve rapidly with the amount of data. But minimizing the total energy via gradient descent requires the calculation 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[23]. This heavily influences the gradient to an extent that it becomes unusable without further processing[18]. There have been several suggestions to remedy this problem but all of them share a significant loss in accuracy compared to $T_s[n][19, 20, 24]$.

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[25]. 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:

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

Here, each density grid point is associated with a group of model weights $\beta$. 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

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

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 smooth 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. ??):

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

Here, for each basis function coefficient, $\lambda^{(l)}$ 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, 26]). This ML-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 functional $F$ and the error in the density $\hat{n}(r)$ on the resulting error in the total energy of any approximate, self-consistent DFT calculation[27]. Let $\tilde{F}$ be an approximation of the many body functional $F$, and $\hat{n}(r)$ the approximate ground-state density when $\tilde{F}$ is used in the Euler equation. Defining $\tilde{E}[n] = \tilde{F}[n] + \int d^3r n(r)v(r)$ yields

$$\Delta E = \tilde{E}[\hat{n}] - E[n] = \Delta E_F + \Delta E_D \quad (4)$$

where $\Delta E_F = \tilde{F}[\hat{n}] - F[n]$ is the functional-driven error, while $\Delta E_D = \tilde{E}[\hat{n}] - \tilde{E}[n]$ is the density-driven error. In most DFT calculations, $\Delta E$ is dominated by $\Delta E_F$. The standard DFT approximations can, in some specific cases, produce abnormally large density errors that dominate the total error. In such situations, using a more
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 the energy error depends on $M$ for ML-OF and ML-HK with different basis sets for the 1-D problem. **b.** Bottom. Errors of the PBE energies (relative to exact values) and the ML maps (relative to PBE) as a function of interatomic spacing, $R$, for H$_2$ with $M = 7$. **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.

Accurate density can greatly improve the result [27–29]. 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. [18] 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 one electron was solved extremely precisely. Up to 200 cases were used to train an ML model $T_s[n]$ for the non-interacting kinetic energy functional $T_s[n]$ via Kernel Ridge Regression (for details, see supplement).

To measure the accuracy of an approximate HK map, the analysis of the previous section is applied to the KS DFT problem. Here $F$ is just $T_s$, the non-interacting kinetic energy, and

$$\Delta E_F = \tilde{T}_s[n] - T_s[n],$$

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, when the density is found from the functional derivative. This method works by following a gradient descent of the
total energy functional based on the gradient of the ML model $T_{s}^{\text{ML}}$,

$$n^{(j+1)} = n^{(j)} - \epsilon P \left( n^{(j)} \right) \frac{\delta}{\delta n} E^{\text{ML}}(n^{(j)}), \quad (6)$$

where $\epsilon$ is a small number and $P(n^{(j)})$ is a localized PCA projection to de-noise the gradient. Here and for all further 1-D results we use

$$E^{\text{ML}}[n] = T_{s}^{\text{ML}}[n] + \int dx \; n(x) \; v(x). \quad (7)$$

The density-driven contribution to the error $\Delta E_{D}$, which we calculate exactly here using the von Weizsäcker kinetic energy[30] is always comparable to, or greater than, the functional-driven error $\Delta E_{F}$, due to the poor quality of the ML functional derivative[18]. The calculation is abnormal, and can be greatly improved by using a more accurate density from a finer grid. As the number of training points $M$ 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 of $T_{s}$ to find the density by solving the Euler equation.

The next set of columns analyzes the ML-HK approach, using a grid basis. The left-most of these columns shows the energy error we obtain by utilizing the ML-HK map:

$$\Delta E = \left| E^{\text{ML}}[n^{\text{ML}}[u]] - E \right|.$$

Note that both ML models, $T_{s}^{\text{ML}}$ and $n^{\text{ML}}$, have been trained using the same set of $M$ training points.

The ML-HK approach is always more accurate than ML-OF, and its relative performance improves as $M$ increases. The next column reports the density-driven error $\Delta E_{D}$ which is an order-of-magnitude smaller than for ML-OF. Lastly, we list an estimate to the density-driven error

$$\Delta E_{D}^{\text{ML}} = \left| E^{\text{ML}}[n^{\text{ML}}[u]] - E^{\text{ML}}[n] \right|,$$

which uses the ML model $T_{s}^{\text{ML}}$ for the kinetic energy functional in 1-D. This proxy is generally a considerable overestimate (a factor of 3 too large), so that the true $\Delta E_{D}$ is always significantly smaller. We use it in subsequent calculations (where we cannot calculate $T_{s}^{\text{ML}}$) to (over-)estimate the energy error due to the HK-ML map.

The last set of columns are density-driven errors for other basis sets. Three variants of the ML-HK map were tested. First, direct prediction of the grid coefficients: In this case, $u_{l}^{(i)} = n_{i}(x_{l})$, $l = 1, \ldots, G$. 500 grid points were used, as in Snyder et al. [18]. This variant is tested in 1-D only; in 3-D the high dimensionality will be prohibitive. Second, a common Fourier basis is tested. The density can be transformed efficiently via the discrete Fourier transform, using 200 Fourier basis functions in total. In 3-D these basis functions correspond to plane waves. The back-projection $u \mapsto n$ to input space is simple, but although the basis functions are physically motivated, they are very general and not specifically tailored to density functions. The performance is almost identical to the grid on average, although maximum errors are much less. For $M = 20$, the error that originates from the basis representation starts to dominate. This is a motivation for exploring, third, a Kernel PCA (KPCA) basis[31]. KPCA[32] 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.

### Molecules

We next apply the ML-HK approach to predict electron densities and energies for a series of small molecules. We test the ML models on KS-DFT results obtained

| $M$ | $\Delta E$ | $\Delta E_{F}$ | $\Delta E_{D}$ | $\Delta E$ | $\Delta E_{D}$ | $\Delta E_{D}^{\text{ML}}$ | $\Delta E_{D}^{\text{KPCA}}$ |
|-----|------------|----------------|---------------|------------|----------------|-----------------|----------------|
| 20  | 7.7        | 47             | 7.7           | 88.8       | 88            | 0.76            | 8.9            |
| 50  | 1.6        | 30             | 1.3           | 7.3        | 7.3           | 0.079           | 0.92           |
| 100 | 0.74       | 17             | 0.2           | 2.6        | 2.6           | 0.027           | 0.43           |
| 200 | 0.17       | 2.9            | 0.039         | 0.6        | 0.17          | 0.042           | 0.59           |

Table I. Energy errors in kcal/mol for the 1-D data set for various $M$, the number of training points. For definitions, see text.
using the PBE exchange-correlation functional[33] and atomic pseudopotentials with the projector augmented wave (PAW) method[34, 35] in the Quantum ESPRESSO code.[36] Since the ML-OF approach applied in the previous section becomes prohibitively expensive in 3-D due to the poor convergence of the gradient descent procedure, we compare the ML-HK map to the ML-KS approach. This approach models the energy directly as a functional of \( v(\mathbf{r}) \), i.e. it trains a model

\[
E^{\text{ML}}[v] = \sum_{i=1}^{M} \alpha_i k(v_i, v)
\]

(10)

via KRR (for details, see supplement). We also apply the ML-HK map with Fourier basis functions and learn an \( E^{\text{ML}}[n] \) model

\[
E^{\text{ML}}[n] = \sum_{i=1}^{M} \alpha_i k(n_i, n)
\]

(11)

instead of \( T^{\text{ML}}[n] \), which avoids implementing the potential energy and exchange-correlation functionals.

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

\[
v(\mathbf{r}) = \sum_{\alpha=1}^{N_a} Z_\alpha \exp \left( -\frac{||\mathbf{r} - R_\alpha||^2}{2\gamma^2} \right)
\]

(12)

where \( R_\alpha \) are the positions and \( Z_\alpha \) are the nuclear charges of the \( N_a \) atoms. The Gaussians potential is used for the ML representation only. The width \( \gamma \) is a hyper-parameter 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[37]. The Gaussians potential is discretized on a coarse grid with grid spacing \( \Delta = 0.08 \) Å. To use the discretized potential in the Gaussian kernel, we flatten it into a vector form and thus use a tensor Frobenius norm.

Our first molecular prototype is \( \text{H}_2 \), with the only degree of freedom, \( R \), denoting the distance between the \( \text{H} \) atoms. A dataset of 150 geometries is created by varying \( R \) between 0.5 and 1.5 Å (sampled uniformly). A randomly chosen subset of 50 geometries are designated as the test set and are unseen by the ML algorithms. These geometries are used to measure the out-of-sample error reported below.

The remaining 100 geometries make up the 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 \( E^\text{ML}[v] \) and \( E^\text{ML}[n] \) models, respectively. Because the required training subsets are so small, careful selection of a subset that covers the complete range of \( R \) is necessary. This is accomplished 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).

For practical applications, it is not necessary to run DFT calculations for the complete grand training set, only for the \( M \) selected training points. Strategies for sampling the conformer space and selecting the grand training set for molecules with more degrees of freedom are explained for \( \text{H}_2\text{O} \) and MD simulations later on.

The performance of the ML-KS map and ML-HK map is compared by training \( E^\text{ML}[v] \) that maps from the Gaussians potential to total energy and \( n^\text{ML}[v] \) that maps from Gaussians potential to the ground-state density in a three-dimensional Fourier basis representation (\( l = 25 \)). The prediction errors are listed in Table II.

The MAE of the energy evaluated using the ML-HK map is significantly smaller than that of the ML-KS map. This indicates that even for a 3-D system, learning the potential-density relationship via the HK map is much easier than directly learning the potential-energy relationship via the KS map.

Fig. 1b shows the errors made by the ML-KS and the ML-HK maps. The error of the ML-HK map is smoother than the ML-KS error and is much smaller, even for the most problematic region when \( R \) is smaller than the equilibrium bond distance of \( R_0 = 0.74 \) Å. The MAE that is introduced by the PBE approximation on the \( \text{H}_2 \) dataset is 2.3 kcal/mol (compared to exact CI calculations), i.e., well above the errors of the ML model and verifies that the error introduced by the ML-HK map is negligible for a DFT calculation.

The next molecular example is \( \text{H}_2\text{O} \), parametrized with three degrees of freedom: two bond lengths and a bond angle. To create a conformer dataset, the optimized structure (\( R_0 = 0.97 \) Å, \( \theta_0 = 104.2^\circ \) using PBE) is taken as a starting point. A total of 350 geometries are then generated by changing each bond length by a uniformly sampled value between ±0.075 Å and varying the angle \( \theta \) between ±8.59 degrees (±0.15 rad) away from \( \theta_0 \) (see supplement for a visualization of the sampled range). For this molecule, the out-of-sample test set again comprises a random subset of 50 geometries, with the remaining 300 geometries used as the grand training set. Because there are now three parameters, it is more difficult to select equidistant samples for the training subset of \( M \) data points. We therefore use a K-means approach to find \( M \) clusters and select the grand training set geometry closest to each cluster’s center for the training subset (see supplement for details).

Models are trained as for \( \text{H}_2 \). The results are given in Table II. As expected, the increase in degrees of freedom for \( \text{H}_2\text{O} \) compared to \( \text{H}_2 \) requires a larger training set size.
| Molecule | $\Delta E$ | $\Delta R_0$ | $\Delta \theta_0$ | $\Delta E_{\text{ML}}$ | $\Delta R_0$ | $\Delta \theta_0$ |
|----------|----------------|----------------|-------------------|---------------------|----------------|----------------|
| $\text{H}_2$ | 5 | 1.3 | 4.3 | 2.2 | — | 0.70 | 2.9 | 0.18 | 0.54 | 1.1 | — |
| | 7 | 0.37 | 1.4 | 0.23 | — | 0.17 | 0.73 | 0.054 | 0.16 | 0.19 | — |
| | 10 | 0.080 | 0.41 | 0.23 | — | 0.019 | 0.11 | 0.017 | 0.086 | 0.073 | — |
| $\text{H}_2\text{O}$ | 5 | 1.4 | 5.0 | 2.1 | 2.2 | 1.1 | 4.9 | 0.056 | 0.17 | 2.3 | 3.8 |
| | 10 | 0.27 | 0.93 | 0.63 | 1.9 | 0.12 | 0.39 | 0.099 | 0.59 | 0.12 | 0.38 |
| | 15 | 0.12 | 0.47 | 0.19 | 0.41 | 0.043 | 0.25 | 0.029 | 0.14 | 0.064 | 0.23 |
| | 20 | 0.015 | 0.064 | 0.043 | 0.16 | 0.0091 | 0.060 | 0.011 | 0.058 | 0.024 | 0.066 |

Table II. Prediction errors on $\text{H}_2$ and $\text{H}_2\text{O}$ with increasing number of training points $M$ for the ML-KS and ML-HK approaches. In addition, the estimated density-driven contribution to the error for the ML-HK approach (Eq. 9) is given. Energies in kcal/mol, bond-lengths in pm, and angles in degrees.

For larger molecules, generating random conformers that sample the full configurational space becomes difficult. Therefore, we next demonstrate that molecular dynamics (MD) using a classical force field can also be used to create the grand training set. As an example, we use benzene ($\text{C}_6\text{H}_6$) with only small fluctuations in atomic positions out of the molecular plane. Appropriate conformers are generated via isothermal MD simulations at 300 K, 350 K, and 400 K using the General Amber Force Field (GAFF)[39] in the PINY_MD package[40]. Saving snapshots from the MD trajectories generates a large set of geometries that are sampled using the K-means approach to obtain 2,000 representative points for the grand training set. Training $n^{\text{ML}}[\varepsilon]$ and $E^{\text{ML}}[\eta]$ is performed as above by running DFT calculations on $M = 2000$ points. We find that the ML error is reduced by creating the training set from trajectories at both the target temperature and a higher temperature to increase the representation of more distorted geometries. The final ML model is tested on 200 conformational snapshots taken from an independent MD trajectory at 300 K (see Fig. 3a). The MAE of the ML-HK map for this data set using training geometries from 300 K and 350 K trajectories is only 0.37 kcal/mol for an energy range that spans more than 10 kcal/mol (see Table III).

For benzene, we further quantify the precision of the ML-HK map in reproducing PBE densities. In Fig. 4, it is clear that the errors in the Fourier basis representation are larger than the errors introduced by the ML-HK map by two orders of magnitude. Furthermore, the ML-HK

Figure 2. **Top.** Distribution of energy errors against PBE on the $\text{H}_2\text{O}$ dataset for ML-KS and ML-HK. The errors are plotted on a symmetric log scale with linear threshold of 0.01, using nearest neighbor interpolation from a grid scan for coloring. Black dots mark the test set geometries with averaged bond lengths. **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 for symmetric geometries ($R$ versus $\theta$). All models trained on $M = 15$ training points. Energies and errors in kcal/mol. A black cross marks the PBE equilibrium position.

$M$. However, even for the more complicated molecule, the ML-HK map is consistently more precise than the ML-KS map, and provides an improved potential energy surface, as shown in Fig. 2. With an MAE of 1.2 kcal/mol for PBE energies relative to CCSD(T) calculations for this data set, we again show that ML does not introduce a new significant source of error.

The ML maps can also be used to find the minimum energy 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[38], which requires a starting point and an evaluation function to be minimized. For the $\text{H}_2\text{O}$ case, the search is restricted to symmetric configurations, with a random symmetric geometry used as the starting point. Results are reported in Table II. The optimizations consistently converge 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 introducing artificial local minima.

For larger molecules, generating random conformers that sample the full configurational space becomes difficult. Therefore, we next demonstrate that molecular dynamics (MD) using a classical force field can also be used to create the grand training set. As an example, we use benzene ($\text{C}_6\text{H}_6$) with only small fluctuations in atomic positions out of the molecular plane. Appropriate conformers are generated via isothermal MD simulations at 300 K, 350 K, and 400 K using the General Amber Force Field (GAFF)[39] in the PINY_MD package[40]. Saving snapshots from the MD trajectories generates a large set of geometries that are sampled using the K-means approach to obtain 2,000 representative points for the grand training set. Training $n^{\text{ML}}[\varepsilon]$ and $E^{\text{ML}}[\eta]$ is performed as above by running DFT calculations on $M = 2000$ points. We find that the ML error is reduced by creating the training set from trajectories at both the target temperature and a higher temperature to increase the representation of more distorted geometries. The final ML model is tested on 200 conformational snapshots taken from an independent MD trajectory at 300 K (see Fig. 3a). The MAE of the ML-HK map for this data set using training geometries from 300 K and 350 K trajectories is only 0.37 kcal/mol for an energy range that spans more than 10 kcal/mol (see Table III).

For benzene, we further quantify the precision of the ML-HK map in reproducing PBE densities. In Fig. 4, it is clear that the errors in the Fourier basis representation are larger than the errors introduced by the ML-HK map by two orders of magnitude. Furthermore, the ML-HK
| Molecule       | Training trajectories | $\Delta E$ MAE max | $\Delta E_{ML}$ MAE max |
|---------------|-----------------------|--------------------|-------------------------|
| Benzene       | 300K                  | 0.42 1.7           | 0.32 1.5                |
|               | 300K + 350K           | 0.37 1.8           | 0.28 1.5                |
|               | 300K + 400K           | 0.47 2.3           | 0.30 1.8                |
| Ethane        | 300K                  | 0.20 1.5           | 0.17 1.3                |
|               | 300K + 350K           | 0.23 1.4           | 0.19 1.1                |
|               | 300K + 400K           | 0.14 1.7           | 0.098 0.62              |
| Malonaldehyde | 300K + 350K           | 0.27 1.2           | 0.21 0.74               |

Table III. Energy and density-driven errors (kcal/mol) of the ML-HK approach on the MD datasets for different training trajectory combinations.

errors in density (as evaluated on a grid in the molecular plane of benzene) are also considerably smaller than the difference in density between density functionals (PBE versus LDA[41]). This result verifies that the ML-HK map is specific to the density used to train the model and should be able to differentiate between densities generated with other electronic structure approaches.

Ethane ($C_2H_6$), with a small energy barrier for the relative rotation of the methyl groups, is also evaluated in the same way. Using geometries sampled using K-means from 300 K and 350 K classical trajectories, the ML-HK model reproduces the energy of conformers with a MAE of 0.23 kcal/mol for an independent MD trajectory at 300 K (Fig. 3b). This test set includes conformers from the sparsely-sampled eclipsed configuration (see supplement). Using points from a 400 K trajectory improves the ML-HK map due to the increased probability of higher energy rotamers in the training set (see Table III). The training set could also be constructed by including explicit rotational conformers, as is common for fitting classical force field parameters[39]. In either case, generating appropriate conformers for training via computationally cheap classical MD significantly decreases the cost of the ML-HK approach.

As an additional proof of the versatility of the ML-HK map, we show that this approach is also able to interpolate energies during an intramolecular proton transfer event in the enol form of malonaldehyde ($C_3H_4O_2$). Classical MD trajectories are run for each tautomer separately, with a fixed bonding scheme, then combined for K-means sampling to create the grand training set. The training set also includes an artificially constructed geometry that is the average of tautomer atomic positions. For the test set, we use snapshots from a computationally expensive Born-Oppenheimer ab initio MD trajectory. Fig. 3c shows that the ML-HK map is able to predict DFT energies during a proton transfer event (MAE of 0.27 kcal/mol) despite being trained on classical geometries that did not include these intermediate points.

**DISCUSSION**

For several decades, density functional theory has been a cross-disciplinary area between theoretical physics, chemistry, and materials sciences. The methods of each field cross-fertilize advances in 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 1-D, this is the first demonstration in 3-D, using real molecules and production-level codes. We also demonstrate that molecular conformers used in the training set can be generated by a range of methods, including informed scans and classical MD simulations. 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 arises in 3-D. Our 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[42]. Following a direct prediction approach with the ML-HK map increases the accuracy consistently on both 1-D examples and 3-D molecules. We are also able to learn density models that outperform energy models trained on much more 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., [25]), but the present work quantifies this observation using standard statistical methods. As the ML-HK map accurately reflects the training densities, more exact methods could also be used to generate the training set densities for functional development.

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 practitioners 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)
Figure 3. Energy errors of ML-HK along MD trajectories. PBE values in blue, ML-HK values in red. a. A 2 ps classical trajectory of benzene. b. A 2 ps classical trajectory of ethane. c. A 0.25 ps ab initio trajectory of malonaldehyde. The ML model correctly predicts energies during a proton transfer in frames 7 to 15 without explicitly including these geometries in the training set.

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[21]. If machine-learning corrections require less data, the method becomes more powerful by taking advantage of existing successes. Furthermore, existing theorems, such as the viral theorem[43], might also be used to directly construct the kinetic energy functional from an ML-HK map. 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.

We also note that, for all the 3-D calculations shown here, we machine-learned \( E[n] \), the entire energy (not just the kinetic energy), which includes some density-functional approximation for XC. But, with a quantum chemical code, we could have trained on much more accurate quantum chemical densities and energies. Thus, the ML-HK maps in principle allow the construction of (nearly) exact density functionals for molecular systems, with the potential to significantly reduce the computational cost of quantum chemistry based MD simulations. All this provides useful directions in which to expand on the results shown here.
Figure 4. The precision of our density predictions using the Fourier basis for ML-HK for the molecular plane of benzene. The plots show a. the difference between the valence density of benzene when using PBE and LDA functionals at the PBE optimized geometry. b. error introduced by using the Fourier basis representation. c. error introduced by the \( n^{\text{ML}}[v] \) density fitting (a.–c. on same color scale). d. the total PBE valence density e. the density differences along a 1-D cut of a.–c. f. the density error introduced with the ML-HK map (same data, but different scale, as in c.).

**METHODS**

Kohn-Sham Density Functional Theory (KS-DFT) 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[44] 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, 3-D DFT calculations for ML models are performed with the Quantum ESPRESSO code[36] using the PBE exchange-correlation functional[33] and projector augmented waves (PAWs)[34, 35] with Troullier-Martin pseudization for describing the ionic cores[45]. All molecules are simulated in a cubic box (L = 20 bohr) with a wave function cutoff of 90 Ry. The 1-D dataset is taken from Snyder et al. [18].

Kernel Ridge Regression (KRR)[46, 47] is a machine learning method for regression. It is a kernelized version of Ridge Regression which minimizes the least squares error and applies an \( \ell_2 \) (Tikhonov) regularization. Let \( x_1, \ldots, x_m \in \mathbb{R}^d \) be the training data points and let \( Y = (y_1, \ldots, y_m)^T \) be their respective labels. KRR then optimizes

\[
\min_{\alpha} \sum_{i=1}^{m} \left| y_i - \sum_{j=1}^{m} \alpha_j k(x_i, x_j) \right|^2 + \lambda \alpha^T K \alpha
\]

where \( k \) is the kernel function and \( \lambda \) is a regularization parameter. \( K \) is the kernel matrix with \( K_{ij} = k(x_i, x_j) \). It admits an analytical solution

\[
\alpha = (K - \lambda I)^{-1} Y.
\]
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\cite{26}.

For the ML-HK map, the canonical error is given by the $L_2$ distance between predicted and true densities

$$ e(\beta) = \sum_{i=1}^{M} \| n_i - n_{ML}^{\beta}[v_i] \|_{L_2} $$

$$ = \sum_{i=1}^{M} \left\| n_i - \sum_{l=1}^{L} \sum_{j=1}^{M} \beta_l^{(i)} k(v_i, v_j) \phi_l \right\|_{L_2}. $$

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

$$ \beta_l^{(i)} = \left( K_{\sigma(i)} + \lambda_l^{(i)} I \right)^{-1} u_l^{(i)}, \quad l = 1, \ldots, L. $$

**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. \cite{12}). Once the model is fixed after training, it is applied unchanged out-of-sample.

**Exact calculations.** Relative energy errors of the ML models trained on KS-DFT calculations are determined by comparing to accurate energies from the MOLPRO Quantum Chemistry Software\cite{48} using the Full Configuration Interaction method for H$_2$ and CCSD(T)\cite{49} for H$_2$O.

**Molecular Dynamics (MD).** For benzene, ethane, and malonaldehyde, GAFF parameters\cite{39} were assigned using the AmberTools package\cite{50}. Geometry optimizations were performed using MP2/6-31g(d) in Gaussian09\cite{51}. Atomic charge assignments are from RESP fits to HF/6-31g(d) calculations at optimized geometries\cite{52} and two additional rotational conformers for ethane.

For the three larger molecules, classical isothermal MD simulations were run using the PINY_MD package\cite{40} with massive Nosé-Hoover chain (NHC) thermostats\cite{53} for atomic degrees of freedom (length = 4, $\tau$ = 20 fs, Suzuki-Yoshida order = 7, multiple time step = 4) and a time step of 1 fs. The r-RESPA multiple time step approach\cite{54} was employed to compute rapidly varying forces more frequently (torsions every 0.5 fs, bonds/bends every 0.1 fs). Systems were equilibrated for 100 ps before collecting snapshots every 100 fs from 1 ns trajectories. Snapshots were aligned to a reference molecule prior to subtracting from the total.

For malonaldehyde, an additional Born-Oppenheimer MD simulation using DFT was run using the QUICK-STEP package\cite{55} in CP2K v. 2.6.2\cite{56}. The PBE exchange-correlation functional\cite{33} was used in the Gaussian and plane wave (GPW) scheme\cite{57} with DZVP-MOLOPT-GTH (m-DZVP) basis sets\cite{58} paired with the appropriate dual-space GTH pseudopotentials\cite{59} optimized for the PBE functional\cite{60}. Wave functions were converged to 1E-7 Hartree using the orbital transformation method\cite{61} on a multiple grid ($n = 5$) with a cutoff of 900 Ry for the system in a cubic box (L = 20 bohr). A temperature of 300 K was maintained using massive NHC thermostats\cite{53} (length = 4, $\tau$ = 10 fs, Suzuki-Yoshida order = 7, multiple time step = 4) and a time step of 0.5 fs.

**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. KR and FB thank the Einstein Foundation for generously funding the ETERNAL project. Work at NYU supported by the U.S. Army Research Office under contract/grant number W911NF-13-1-0387 (MET and LV). Ab initio trajectory was run using High Performance Computing resources at NYU. Other DFT simulations were run using High Performance Computing resources at MPI Halle.

**AUTHOR CONTRIBUTIONS**

FB performed DFT calculations and ML experiments. LV performed classical and ab initio MD simulations. 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.

---

\cite*{1} W. Kohn and L. J. Sham, Phys. Rev. 140, A1133 (1965).

\cite{2} A. Pribram-Jones, D. A. Gross, and K. Burke, Annual Review of Physical Chemistry 66, 283 (2015).

\cite{3} J. P. Perdew, Phys. Rev. B 33, 8822 (1986).

\cite{4} A. D. Becke, The Journal of Chemical Physics 98, 5648 (1993).

\cite{5} “Materials genome initiative for global competitiveness,” (2011).

\cite{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).
[55] J. VandeVondele, M. Krack, F. Mohamed, M. Parrinello, T. Chassaing, and J. Hutter, Computer Physics Communications 167, 103 (2005).

[56] J. Hutter, M. Iannuzzi, F. Schiffmann, and J. VandeVondele, Wiley Interdisciplinary Reviews: Computational Molecular Science 4, 15 (2014).

[57] G. Lippert, J. Hutter, and M. Parrinello, Molecular Physics 92, 477 (2010).

[58] J. VandeVondele and J. Hutter, The Journal of Chemical Physics 127, 114105 (2007).

[59] S. Goedecker, M. Teter, and J. Hutter, Phys. Rev. B 54, 1703 (1996).

[60] M. Krack, Theoretica Chimica Acta 114, 145 (2005).

[61] J. VandeVondele and J. Hutter, The Journal of Chemical Physics 118, 4365 (2003).
By-passing the Kohn-Sham equations with machine learning
Supplemental Information
Felix Brockherde,1, 2 Leslie Vogt,3 Li Li,4 Mark E. Tuckerman,3, 5, 6 Kieron Burke,7, 4, ∗ and Klaus-Robert Müller1, 8, ∗
1 Machine Learning Group, Technische Universität Berlin, Marchstr. 23, 10587 Berlin, Germany
2 Max-Planck-Institut für Mikrostrukturphysik, Weinberg 2, 06120 Halle, Germany
3 Department of Chemistry, New York University, New York, NY 10003, USA
4 Departments of Physics and Astronomy, University of California, Irvine, CA 92697, USA
5 Courant Institute of Mathematical Science, New York University, New York, NY 10003, USA
6 NYU-ECNU Center for Computational Chemistry at NYU Shanghai, 3663 Zhongshan Road North, Shanghai 200062, China
7 Departments of Chemistry, University of California, Irvine, CA 92697, USA
8 Department of Brain and Cognitive Engineering, Korea University, Anam-dong, Seongbuk-gu, Seoul 136–713, Republic of Korea

(Dated: March 3, 2017)

KERNEL RIDGE REGRESSION

Kernel Ridge Regression[1, 2] (KRR) is a machine learning method for regression. We introduce the method for abstract training points \((x_i, y_i)\), i.e. features \(x_1, \ldots, x_M \in \mathbb{R}^d\) and associated labels \(Y = (y_1, \ldots, y_M)^T \in \mathbb{R}^M\) and describe the actual models used in the main text afterwards. We want to model a function \(f : \mathbb{R}^d \to \mathbb{R}\) that maps from features to labels. This model should not be `learned by heart’ but perform well on unseen data (i.e. generalize). We first restrict the set of possible functions to the reproducing kernel Hilbert space (RKHS) \(\mathcal{H}\) on the space of discretized densities that is induced by the Gaussian kernel function

\[
k(x, x') = \exp \left( -\frac{||x - x'||^2}{2\sigma^2} \right). \tag{1}
\]

The restriction is very mild and rather technical; more interesting is the choice of the kernel function which determines the scalar product (and thus the norm) of the RKHS. Leaving rigor aside, the Gaussian kernel induces an RKHS norm \(||f||_\mathcal{H}\) that is smaller for simpler, smoother functions and higher for more complicated, oscillating functions. We minimize the empirical risk functional

\[
\mathcal{C}(f) = \sum_{i=1}^{M} |y_i - f(x_i)|^2 + \lambda \|f\|_\mathcal{H}^2 \tag{2}
\]

that defines a trade-off between error on the training points and smoothness of the function controlled by the hyper-parameter \(\lambda\).

The representer theorem[3] allows us to assume that the solution to Eq. 2 is given by a linear combination of kernel functions \(f = \sum_{i=1}^{M} \alpha_i k(x_i, \cdot)\). It now suffices to solve

\[
\mathcal{C}(\alpha) = \sum_{i=1}^{M} |y_i - f(x_i)|^2 + \lambda \|f\|_\mathcal{H}^2 = \sum_{i=1}^{M} |y_i - f(x_i)|^2 + \lambda \alpha^T K \alpha, \tag{4}
\]

where \(K_{ij} = k(x_i, x_j)\) is the kernel matrix. The solution is given by

\[
\alpha = (K - \lambda I)^{-1} Y. \tag{5}
\]

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. [4]). Once the model is fixed after training, it is applied unchanged out-of-sample.

We use this method for various maps:

- Non-interacting kinetic energy functional \((T_n^{ML}[n], 1-D)\). The training points are given by pairs of densities and associated kinetic energies. We discretize the densities and use them in vectorial form, i.e. \(n \in \mathbb{R}^G\). Thus, the functional \(\mathcal{L}^2 \to \mathbb{R}\) is modeled as a function \(\mathbb{R}^G \to \mathbb{R}\)
- ML-OF map (1-D). The training points are given by pairs of discretized 1-D box potentials and associated total energies.
- ML-KS map (3-D). The training points are given by pairs of discretized Gaussians potentials (as described in the main text) and total energies.
- Total energy functional \((E_{T_s}^{ML}[n], 3-D)\). The training points are given by pairs of densities in basis function representation (see below) and associated total energies. Just as for \(T_s^{ML}\), this functional is modeled as a function.

ML HOHENBERG-KOHN MAP

The basis representation for the densities is given by
\[ n(x) = \sum_{l=1}^{L} u^{(l)} \phi_l(x), \quad (6) \]

where \( \phi_l \) are the \( L \) basis functions. We introduce some notation and continue to write the density in grid representation as \( n \), and its basis coefficients as \( u \). We can then write the HK map model as

\[ n^{ML}[v](x) = \sum_{l=1}^{L} u^{(l)}[v] \phi_l(x), \quad (7) \]

where the \( L \) basis function coefficients are regular KRR models.

\[ u^{(l)}[v] = \sum_{i=1}^{M} \beta_i^{(l)} k(v, v_i), \quad (8) \]

of external potentials \( v \) with a Gaussian kernel function. The contribution of the error to the cost function can be formulated as

\[ e(\beta) = \sum_{i=1}^{M} \| n_i - n^{ML}[v_i] \|_{L_2}^2 \]

\[ = \sum_{i=1}^{M} \left\| n_i - \sum_{l=1}^{L} \sum_{j=1}^{M} \beta_j^{(l)} k(v_i, v_j) \phi_l \right\|_{L_2}, \quad (10) \]

with the \( L_2 \) norm. We write this cost function in terms of basis function coefficients. This can be viewed as projecting the inside of the norm on each basis function. Assuming orthogonality of the basis functions yields

\[ e(\beta) = \sum_{i=1}^{M} \sum_{l=1}^{L} \left| u_i^{(l)} - \sum_{j=1}^{M} \beta_j^{(l)} k(v_i, v_j) \right|^2. \quad (11) \]

where \( u_i^{(l)} = \langle n_i, \phi_l \rangle \) is the \( l \)-th basis function coefficient of the \( i \)-th training density, as defined in Eq. 6 if orthogonality is satisfied. After reordering the sums over \( i \) and \( l \), we view each \( l \) independently and solve analogously to regular KRR

\[ \beta^{(l)} = \left( K_{\sigma^{(l)}} + \lambda^{(l)} I \right)^{-1} u^{(l)}, \quad l = 1, \ldots, L \quad (12) \]

where, for each basis function \( l \), \( \lambda^{(l)} \) is a regularization parameter, \( K_{\sigma^{(l)}} \) 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 [4, 5]).

**BASIS FUNCTIONS**

**Fourier basis.** We define the basis as

\[ \phi_l(x) = \begin{cases} \cos \left( \frac{2\pi x(l-1)}{2} \right), & l \text{ odd} \\ \sin \left( \frac{2\pi x l}{2} \right), & l \text{ even} \end{cases} \]

\[ l = 1, \ldots, L. \quad (13) \]

We transform the density efficiently via the discrete Fourier transform

\[ u^{(l)}_i = \sum_{m=1}^{G} n_i(x_m) \phi_l(x_m). \quad (14) \]

The back-projection is written as

\[ n^{ML}[v](x) = \sum_{l=1}^{L} u^{(l)}[v] \phi_l(x). \quad (15) \]

**KPCA basis.** We define the basis as:

\[ \phi^{KPCA}_l = \sum_{j=1}^{M} p_j^{(l)} \Phi(n_j). \quad (16) \]

The parameters \( p_j^{(l)} \) are found by eigen-decomposition of the Kernel matrix. The KCPA basis coefficients are given by

\[ u_i^{(l)} = \langle \Phi(n_i), \phi^{KPCA}_l \rangle = \sum_{j=1}^{M} p_j^{(l)} k(n_j, n_i) \quad (17) \]

with kernel map \( \Phi \). The back-projection for KPCA is not trivial but several solutions exist. We follow Bakir et al. [6] and learn the back-projection map.

**GRADIENT DESCENT ISSUES**

There are two ways to remedy problems of the gradient descent procedure: First, the gradient descent step can be “de-noised” by projecting the gradient onto the data manifold and thus removing the noisy directions. Secondly, the directions outside of the data manifold can be removed in a preprocessing step to get rid of the influence of the noisy directions on the gradient completely. Both methods yield similar results.

Several approaches exist for describing and projecting onto the data manifold. Common to each approach is the idea to find principle components and to project on
those in which direction the densities have largest variance. Best results are reported [7] by using Kernel Principle Component Analysis [8] (KPCA), a non-linear generalization of PCA.

There are three issues with the assumed gradient-based approaches: First, the correct choice of the number of (K)PCA components $K$ has to be made. It is generally possible to view it as a hyper-parameter and find the optimal $K$ via cross-validation. However, we can not choose fractional $K$s. One $K$ might be not enough and $K + 1$ too much information. Second, the data points only lie in a bounded region of a manifold that can be described via PCA components. It is still possible for the gradient descent to walk outside this bounded region toward a point where the model has no information and thus the gradients become inaccurate. A (K)PCA method that only accesses the scalar products between points in the data set can not solve this[9]. Third, it might not be possible to find a suitable pre-image for a ground-state density given by (K)PCA coefficients[10].

MOLECULAR DATASETS

The extent of the dataset for H$_2$O is visualized in Fig. 1. In this case, conformers were generated from random displacements from the optimized geometry.

For benzene and ethane, conformers were generated from isothermal molecular dynamics (MD) trajectories. The range of atomic positions from combined 1 ns 300 K and 350 K trajectories is shown in Fig. 2 for benzene and Fig. 3 for ethane after snapshots are aligned to a reference molecule. For malonaldehyde, the classical MD trajectories include 0.5 ns for each tautomer at each temperature. Resulting conformers that are used to create the K-means sampled training set are shown in Fig. 4. The test set is taken from an ab initio MD trajectory at 300 K.

DFT CONVERGENCE

For our 3-D DFT calculations in Quantum Espresso[11], we center a water molecule in a cubic cell and converge three variables: the kinetic energy
cutoff for wavefunctions $\text{ecutwfc}$ in steps of 10 Ry, the kinetic energy cutoff for charge density and potential $\text{ecutrho}$ in steps of 40 Ry, and the cell dimension $\text{celldm}$ in steps of 1 bohr. We increase parameters until increasing any parameter does not change the equilibrium position total energy by more than 0.01 kcal/mol for $\text{H}_2\text{O}$. We end up with $\text{ecutwfc}$ of 90 Ry, $\text{ecutrho}$ of 360 Ry, and $\text{celldm}$ of 20 bohr, which are used for all other molecules in this work.

**SAMPLING**

For $\text{H}_2$, since there is only one atomic distance to adjust, we take the $M$ equi-distant points in the parameter range and for each of these points select the training point that is closest.

For larger molecules with more parameters ($\text{H}_2\text{O}$, Benzene, Ethane, Malonaldehyde) we also want to cover the conformer space in a way that all conformers are relatively close to at least one training point.

Assuming $p_i$ are the parameters of conformer $i$ and $i \in \tilde{P}_j$ if and only if $\tilde{p}_j$ is closest to $p_i$, we want to find $\tilde{p}_j$, $j = 1 \ldots M$ that minimize

$$\sum_{j=1}^{M} \sum_{i \in \tilde{P}_j} ||\tilde{p}_j - p_i||^2.$$  

(18)

K-means[12] solves this problem for continuous $\tilde{p}_j$. However, since K-means returns only locally optimal solutions, we rerun the algorithm 50 times and select the solution which minimizes Eq. 18. We choose the points $p_i$ closest to each $\tilde{p}_j$ as training points.

**LOGIC OF DENSITY FUNCTIONAL THEORY (DFT)**

Within the Born-Oppenheimer approximation in non-relativistic quantum mechanics, and using atomic units, the Hohenberg-Kohn paper[13] laid the theoretical framework of all modern DFT. The first statement is that the mapping

$$v(r) \leftrightarrow n(r)$$  

(19)

is one-to-one, i.e., at most one potential can give rise to a given ground-state energy, even in a quantum many-body problem, for given interaction among particles and statistics (i.e., fermions or bosons). A follow-up claim is that the ground-state energy of an electronic system can be found from

$$E[v] = \min_n \left\{ F[n] + \int d^3r \, n(r) v(r) \right\}$$  

(20)

where $F[n]$ is a density functional containing all many-body effects. The minimizing density is the solution to the Euler equation:

$$\frac{\delta F}{\delta n(r)} + v(r) = \text{const}$$  

(21)

It is the direct map between densities and potentials that we machine-learn in this paper. We call it the HK density map, $n[v](r)$.

The KS scheme avoids direct approximation of $F$ by imagining a fictitious system of non-interacting electrons with the same density as the real one[14]. The KS equations are:

$$\left\{ -\frac{1}{2} \nabla^2 + v_h(r) \right\} \phi_i(r) = \epsilon_i \phi_i(r)$$  

(22)

where $\epsilon_i$ are the KS eigenvalues and $\phi_i$ the KS orbitals.

$$v_h(r) = v(r) + v_H(r) + v_{XC}(r)$$  

(23)

where $v_H(r)$ is the Hartree potential and $v_{XC}(r)$ is the exchange-correlation potential. The true energy of the system is then reconstructed from the self-consistent density $n(r) = \sum_i |\phi_i(r)|^2$ via

$$E[n] = T_s[n] + U[n] + \int d^3r \, n(r) v(r) + E_{XC}[n]$$  

(24)
where $T_s[n]$ is the kinetic energy of the non-interacting electrons and $U[n]$ is the Hartree energy. $E_{XC}[n]$ is the exchange-correlation (XC) energy and implicitly defined by Eq. 24. Most calculations[15] use simple approximations that depend only on the density and its gradient to determine $E_{XC}$, called generalized gradient approximations, or replace a fixed fraction of the approximate exchange with the exact exchange from a Hartree-Fock calculation (called a hybrid). Requiring the XC potential to be the functional derivative of $E_{XC}$ ensures that the self-consistent solution of Eq. 22 minimizes the energy of Eq. 24 for the given $v(r)$ and $E_{XC}[n]$.

* to whom correspondence should be addressed.

[1] 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).

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

[3] B. Schölkopf, R. Herbrich, and A. J. Smola, “A generalized representer theorem,” in *Computational Learning Theory: 14th Annual Conference on Computational Learning Theory, COLT 2001 and 5th European Conference on Computational Learning Theory, EuroCOLT 2001 Amsterdam, The Netherlands, July 16–19, 2001 Proceedings*, edited by D. Helmbold and B. Williamson (Springer Berlin Heidelberg, Berlin, Heidelberg, 2001) pp. 416–426.

[4] 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).

[5] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, IEEE Trans. Neural Netw. 12, 181 (2001).

[6] G. H. Bakir, J. Weston, and B. Schölkopf, in *Advances in Neural Information Processing Systems*, Vol. 16, edited by S. Thrun, L. K. Saul, and B. Schölkopf (MIT Press, 2004) pp. 449–456.

[7] J. C. Snyder, M. Rupp, K.-R. Müller, and K. Burke, Int. J. Quantum Chem. 115, 1102 (2015).

[8] B. Schölkopf, A. Smola, and K.-R. Müller, Neural Computation 10, 1299 (1998).

[9] F. J. Király, M. Kreuzer, and L. Theran, arXiv:1406.2646 [cs, math, stat] (2014).

[10] B. Schölkopf, S. Mika, C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. Smola, IEEE Trans. Neural Netw. 10, 1000 (1999).

[11] 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).

[12] H. Steinhaus, Bull. Acad. Polon. Sci. Cl. III. 4, 801 (1956).

[13] P. Hohenberg and W. Kohn, Phys. Rev. 136, B864 (1964).

[14] W. Kohn and L. J. Sham, Phys. Rev. 140, A1133 (1965).

[15] A. Pribram-Jones, D. A. Gross, and K. Burke, Annual Review of Physical Chemistry 66, 283 (2015).