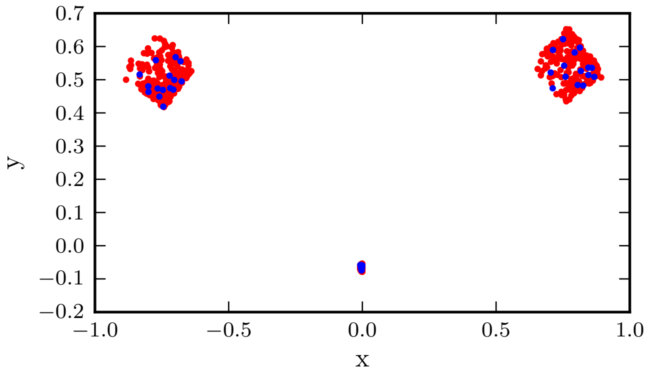
By-passing the Kohn-Sham equations with machine learning
Supplemental Information

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)

Figure 1. The extent of our H_2O dataset. The figure shows the atom coordinates in angstrom. Blue are atoms from 15 training points, red from 200 test points.



H_2O DATASET

The extent of the dataset is visualized in Fig. 1.

SAMPLING

For 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 H_2O, where we have three parameters, we use K-means[1] to find M cluster centers and for each center select the training point that is closest. We repeat K-means 50 times and select the solution with the lowest K-means criterion.

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 [2] by using Kernel Principle Component Analysis[3] (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 Ks. 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[4]. Third, it might not be possible to find a suitable pre-image for a ground-state density given by (K)PCA coefficients[5].

DFT convergence

For our 3-D DFT calculations in Quantum Espresso[6], we center the molecule in a cubic cell and converge three variables: the kinetic energy cutoff for wavefunctions `ecutwfc` in steps of 10 Ry, the kinetic energy cutoff for charge density and potential `ecutrho` in steps of 40 Ry, and the cell dimension `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 either H2 or H2O. We end up with `ecutwfc` of 90 Ry, `ecutrho` of 360 Ry, and `celldm` of 20 bohr.

Kernel Ridge Regression

Kernel Ridge Regression[7, 8] (KRR) is a machine learning method for regression. Using KRR, the non-interacting kinetic energy functional T_s is of the form

$$T_\mathrm{s}^\mathrm{ML}[\mathbf{n}] = \sum_{j=1}^{M} \alpha_i k(\mathbf{n}, \mathbf{n}_j), \tag{1}$$

where $\mathbf{n} = n(x_1), \dots n(x_G)$ is a ground-state density discretized in a grid form, $\mathbf{n}_i$ are the training densities, and k is the Gaussian kernel function,

$$k(\mathbf{n}, \mathbf{n}') = \exp\left(-\frac{||\mathbf{n} - \mathbf{n}'||^2}{2\sigma^2}\right), \tag{2}$$

with the kernel width σ as hyper-parameter. The $\boldsymbol{\alpha}$ parameters are learnt by minimizing the cost function, consisting of squared error and a regularizer that enforces smoothness on the model function,

$$\mathcal{C}(\boldsymbol{\alpha}) = \sum_{i=1}^{m} |T_\mathrm{s}[\mathbf{n}_i] - T_\mathrm{s}^\mathrm{ML}[\mathbf{n}_i]|^2 + \lambda \|\boldsymbol{\alpha}\|^2, \tag{3}$$

where λ is another hyper-parameter and $T_\mathrm{s}[\mathbf{n}]$ is the kinetic energy that corresponds to the ground-state density $\mathbf{n}$. The solution is given by

$$\boldsymbol{\alpha} = (\mathbf{K} - \lambda \mathbf{I})^{-1} \mathbf{T}, \tag{4}$$

where $\mathbf{K}_{ij} = k(\mathbf{n}_i, \mathbf{n}_j)$ and $\mathbf{T} = (T_\mathrm{s}[\mathbf{n}_1], \dots T_\mathrm{s}[\mathbf{n}_m])^\intercal$ are the kinetic energies of the training densities.

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

ML Hohenberg-Kohn map

The basis representation for the densities is given by

$$n(x) = \sum_{l=1}^{L} u^{(l)} \phi_l(x), \tag{5}$$

where ϕ_l are the L basis functions. We introduce some notation and write the density in grid representation as $\mathbf{n}$, and its basis coefficients as $\mathbf{u}$. We can then write the HK map model as

$$n^\mathrm{ML}[v](x) = \sum_{l=1}^{L} u^{(l)}[v] \phi_l(x), \tag{6}$$

where the L basis function coefficients are regular KRR models,

$$u^{(l)}[v] = \sum_{i=1}^{M} \beta_i^{(l)} k(v, v_i), \tag{7}$$

of external potentials $\mathbf{v}$ with a Gaussian kernel function. The cost function can be formulated as

$$C(\boldsymbol{\beta}) = \sum_{i=1}^{M} \|n_i - n^\mathrm{ML}[v_i]\|_{\mathcal{L}_2} \tag{8}$$

$$= \sum_{i=1}^{M} \left\| n_i - \sum_{l=1}^{L} \sum_{j=1}^{M} \beta_j^{(l)} k(v_i, v_j) \phi_l \right\|_{\mathcal{L}_2}, \tag{9}$$

with the $\mathcal{L}_2$ 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

$$C(\boldsymbol{\beta}) = \sum_{i=1}^{M} \sum_{l=1}^{L} \left| u_i^{(l)} - \sum_{j=1}^{M} \beta_j^{(l)} k(v_i, v_j) \right|^2. \tag{10}$$

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. 5 if orthogonality is satisfied. After reordering the sums over i and l, we solve for each l independently in an analytical form analogously to regular KRR

$$\boldsymbol{\beta}^{(l)} = (K_{\sigma^{(l)}} + \lambda^{(l)} I)^{-1} \mathbf{u}^{(l)}, \quad l = 1, \dots, L \tag{11}$$

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 [9, 10]).

Basis functions

Fourier basis. We define the basis as

$$\phi_l(x) = \begin{cases} \cos\{2\pi x(l-1)/2\}, & l \text{ odd} \\ \sin\{2\pi x l/2\}. & l \text{ even} \end{cases} \quad l = 1, \dots, L \tag{12}$$

We transform the density efficiently via the discrete Fourier transform

$$u_i^{(l)} = \sum_{m=1}^{G} n_i(x_m) \phi_l(x_m). \tag{13}$$

The back-projection is written as

$$n^{\mathrm{ML}}[v](x) = \sum_{l=1}^{L} u^{(l)}[v]\phi_l(x). \tag{14}$$

KPCA basis. We define the basis as:

$$\phi_l^{\mathrm{KPCA}} = \sum_{j=1}^{M} p_j^{(l)} \Phi(n_j). \tag{15}$$

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

$$u_i^{(l)} = \langle \Phi(n_i), \phi_l^{\mathrm{KPCA}} \rangle = \sum_{j=1}^{M} p_j^{(l)} k(n_j, n_i) \tag{16}$$

with kernel map Φ. The back-projection for KPCA is not trivial but several solutions exist. We follow Bakir *et al.* [11] and learn the back-projection map.

Logic of Density Functional Theory (DFT)

Within the Born-Oppenheimer approximation in non-relativistic quantum mechanics, and using atomic units, the Hohenberg-Kohn paper[12] laid the theoretical framework of all modern DFT. The first statement is that the mapping

$$v(\mathbf{r}) \longleftrightarrow n(\mathbf{r}) \tag{17}$$

is one-to-one, i.e., at most one potential can give rise to a given ground-state density, 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

$$E[v] = \min_n \left\{ F[n] + \int d^3\mathbf{r}\, n(\mathbf{r}) v(\mathbf{r}) \right\} \tag{18}$$

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

$$\frac{\delta F}{\delta n(\mathbf{r})} + v(\mathbf{r}) = \mathrm{const} \tag{19}$$

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

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[13]. The KS equations are:

$$\left\{ -\frac{1}{2}\nabla^2 + v_{\mathrm{s}}(\mathbf{r}) \right\} \phi_i(\mathbf{r}) = \epsilon_i \phi_i(\mathbf{r}) \tag{20}$$

where ϵ_i are the KS eigenvalues and ϕ_i the KS orbitals.

$$v_{\mathrm{s}}(\mathbf{r}) = v(\mathbf{r}) + v_{\mathrm{H}}(\mathbf{r}) + v_{\mathrm{XC}}(\mathbf{r}) \tag{21}$$

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

$$E[n] = T_{\mathrm{s}}[n] + U[n] + \int d^3\mathbf{r}\, n(\mathbf{r}) v(\mathbf{r}) + E_{\mathrm{XC}}[n] \tag{22}$$

where $T_{\mathrm{s}}[n]$ is the kinetic energy of the non-interacting electrons and $U[n]$ is the Hartree energy. $E_{\mathrm{XC}}[n]$ is the exchange-correlation (XC) energy and implicitly defined by Eq. 22. Most calculations[14] 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. 20 minimizes the energy of Eq. 22 for the given $v(\mathbf{r})$ and $E_{\mathrm{XC}}[n]$.

[*] to whom correspondence should be addressed

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

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

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

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

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

[6] 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).

[7] 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).

[8] 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).

[9] 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).

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

[11] 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.

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

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

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