Finding Density Functionals with Machine Learning

John C. Snyder,[1] Matthias Rupp,[2,3] Katja Hansen,[2] Klaus-Robert Müller,[2,4] and Kieron Burke[1]

[1]*Departments of Chemistry and of Physics, University of California, Irvine, California 92697, USA*
[2]*Machine Learning Group, Technical University of Berlin, Berlin 10587, Germany*
[3]*Institute of Pharmaceutical Sciences, Eidgenössische Technische Hochschule Zürich, Zürich 8093, Switzerland*
[4]*Department of Brain and Cognitive Engineering, Korea University, Anam-dong, Seongbuk-gu, Seoul 136-713, Korea*

(Received 16 December 2011; published 19 June 2012)

Machine learning is used to approximate density functionals. For the model problem of the kinetic energy of noninteracting fermions in 1D, mean absolute errors below 1 kcal/mol on test densities similar to the training set are reached with fewer than 100 training densities. A predictor identifies if a test density is within the interpolation region. Via principal component analysis, a projected functional derivative finds highly accurate self-consistent densities. The challenges for application of our method to real electronic structure problems are discussed.

DOI: 10.1103/PhysRevLett.108.253002 PACS numbers: 31.15.E−, 02.60.Gf, 31.15.X−, 89.20.Ff

Each year, more than 10 000 papers report solutions to electronic structure problems using the Kohn-Sham (KS) density functional theory (DFT) [1,2]. All approximate the exchange-correlation (XC) energy as a functional of the electronic spin densities. The quality of the results crucially depends on these density functional approximations. For example, the present approximations often fail for strongly correlated systems, rendering the methodology useless for some of the most interesting problems.

Thus, there is a never-ending search for improved XC approximations. The original local density approximation (LDA) of Kohn and Sham [2] is uniquely defined by the properties of the uniform gas and has been argued to be a universal limit of all systems [3]. But the refinements that have proven useful in chemistry [4] and materials [5] are not, and they differ both in their derivations and details. Traditionally, physicists favor a nonempirical approach, deriving approximations from quantum mechanics and avoiding fitting to specific finite systems [6]. Such nonempirical functionals can be considered controlled extrapolations that work well across a broad range of systems and properties, bridging the divide between molecules and solids. Chemists typically use a few [7,8] or several dozen [9] parameters to improve the accuracy on a limited class of molecules. Empirical functionals are limited interpolations that are more accurate for the molecular systems they are fitted to, but often fail for solids. Passionate debates are fueled by this cultural divide [10].

Machine learning (ML) is a powerful tool for finding patterns in high-dimensional data. ML employs algorithms by which the computer learns from empirical data via induction, and it has been very successful in many applications [11–13]. In ML, intuition is used to choose the basic mechanism and representation of the data, but not directly applied to the details of the model. Mean errors can be systematically decreased with an increasing number of inputs. In contrast, human-designed empirical approximations employ standard forms derived from general principles, fitting the parameters to training sets. These explore only an infinitesimal fraction of all possible functionals and use relatively few data points.

DFT is useful for electronic structure because the underlying many-body Hamiltonian is simple, while an accurate solution of the Schrödinger equation is very demanding. All electrons Coulomb repel one another and have spin 1/2, which makes the Hohenberg-Kohn theorem [1] possible. But real electronic structure problems are further limited to only those one-body potentials due to Coulomb attraction to the nuclei. ML is a natural tool for taking maximum advantage of this simplicity. For ML to be useful, a pattern must exist, but one that evades human intuition. Furthermore, most present approximations begin from LDA [2] and fail miserably when LDA is a poor starting point. A ML-produced functional suffers no such bias, and so it should be most useful where present approximations fail if it has good examples to train on.

Here, we adapt ML to a prototype density functional problem: noninteracting spinless fermions confined to a 1D box, subject to a smooth potential. We define the key technical concepts that are needed to apply ML to DFT problems. The accuracy we achieve in approximating the kinetic energy (KE) of this system is far beyond the capabilities of any present approximations, and it is even sufficient to produce highly accurate self-consistent densities. Our ML approximation (MLA) achieves chemical accuracy using many more inputs, but requires far less insight into the underlying physics.

We illustrate the accuracy of our MLA with Fig. 1, in which the functional was constructed from 100 densities on a dense grid. This success opens up a new approach to functional approximation, entirely distinct from previous approaches: our MLA contains $\sim 10^5$ empirical numbers and satisfies none of the standard exact conditions.

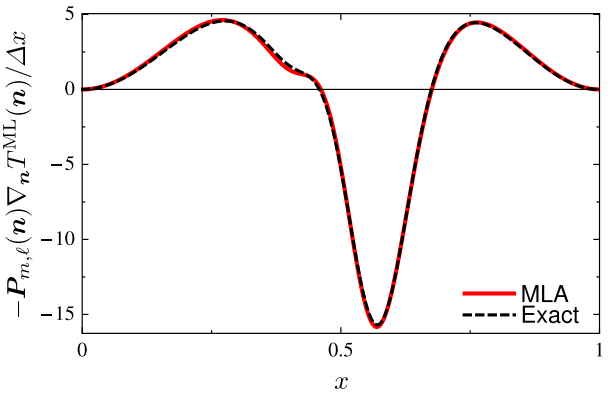

FIG. 1 (color online). Comparison of a projected (see within) functional derivative of our MLA with the exact curve.

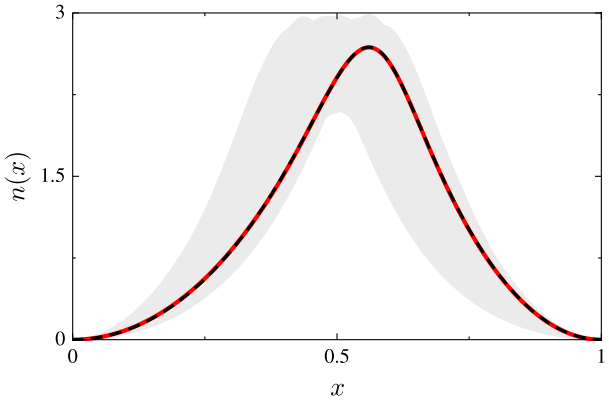

FIG. 2 (color online). The shaded region shows the extent of variation of $n(x)$ within our data set for $N = 1$. Exact (red, solid) and a self-consistent (black, dashed) density for potential of Fig. 3.

The prototype DFT problem we consider is N noninteracting spinless fermions confined to a 1D box, $0 \le x \le 1$, with hard walls. For continuous potentials $v(x)$, we solve the Schrödinger equation numerically with the lowest N orbitals occupied, finding the KE and the electronic density $n(x)$, the sum of the squares of the occupied orbitals. Our aim is to construct a MLA for the KE $T[n]$ that bypasses the need to solve the Schrödinger equation—a 1D analog of orbital-free DFT [14]. (In 3D orbital-free DFT, the local approximation as used in the Thomas-Fermi theory, is typically accurate to within 10%, and the addition of the leading gradient correction reduces the error to about 1% [15]. Even this small an error in the total KE is too large to give accurate chemical properties.)

First, we specify a class of potentials from which we generate densities, which are then discretized on a uniform grid of G points. We use a linear combination of three Gaussian dips with different depths, widths, and centers,

$$v(x) = -\sum_{i=1}^{3} a_i \exp[-(x - b_i)^2/(2c_i^2)]. \tag{1}$$

We generate 2000 such potentials, randomly sampling $1 < a < 10$, $0.4 < b < 0.6$, and $0.03 < c < 0.1$. For each $v_j(x)$, we find for N up to four electrons, the KE $T_{j,N}$ and density $\boldsymbol{n}_{j,N}$ in $\mathbb{R}^G$ on the grid using Numerov's method [16]. For $G = 500$, the error in $T_{j,N}$ due to discretization is less than 1.5×10^{-7}. We take 1000 densities as a test set, and choose M others for training. The variation in this data set for $N = 1$ is illustrated in Fig. 2.

Kernel ridge regression is a nonlinear version of regression with regularization to prevent overfitting [17]. For kernel ridge regression, our MLA takes the form,

$$T^{\mathrm{ML}}(\boldsymbol{n}) = \bar{T} \sum_{j=1}^{M} \alpha_j k(\boldsymbol{n}_j, \boldsymbol{n}), \tag{2}$$

where α_j are weights to be determined, $\boldsymbol{n}_j$ are training densities, and k is the kernel, which measures similarity between densities. Here, $\bar{T}$ is the mean KE of the training set, inserted for convenience. We choose a Gaussian kernel, common in ML,

$$k(\boldsymbol{n}, \boldsymbol{n}') = \exp[-\|\boldsymbol{n} - \boldsymbol{n}'\|^2/(2\sigma^2)], \tag{3}$$

where the hyperparameter σ is called the length scale. The weights are found by minimizing the cost function,

$$\mathcal{C}(\boldsymbol{\alpha}) = \sum_{j=1}^{M} \Delta T_j^2 + \lambda \|\boldsymbol{\alpha}\|^2, \tag{4}$$

where $\Delta T_j = T_j^{\mathrm{ML}} - T_j$ and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_M)$. The second term is a regularizer that penalizes large weights to prevent overfitting. The hyperparameter λ controls regularization strength. Minimizing $\mathcal{C}(\boldsymbol{\alpha})$ gives

$$\boldsymbol{\alpha} = (\boldsymbol{K} + \lambda \boldsymbol{I})^{-1} \boldsymbol{T}, \tag{5}$$

where $\boldsymbol{K}$ is the kernel matrix with elements $\boldsymbol{K}_{ij} = k(\boldsymbol{n}_i, \boldsymbol{n}_j)$, and $\boldsymbol{I}$ is the identity matrix. Then σ and λ are determined through tenfold cross validation: the training set is partitioned into 10 bins of equal size. For each bin, the functional is trained on the remaining samples, and σ and λ are optimized by minimizing the mean absolute error (MAE) on the bin. The partitioning is repeated up to 40 times, and the hyperparameters are chosen as the median over all bins.

Table I gives the performance of T^{ML} [Eq. (2)] trained on M N-electron densities and evaluated on the corresponding test set. The mean KE of the test set for $N = 1$ is 5.40 hartree (3390 kcal/mol). To contrast, the LDA in 1D is $T^{\mathrm{loc}}[n] = \pi^2 \int dx\, n^3(x)/6$ and the von Weizsäcker functional is $T^W[n] = \int dx\, n'(x)^2/[8n(x)]$. For $N = 1$, the MAE of T^{loc} on the test set is 217 kcal/mol, and the modified gradient expansion approximation [19], $T^{\mathrm{MGEA}}[n] = T^{\mathrm{loc}}[n] - cT^W[n]$, has a MAE of 160 kcal/mol, where $c = 0.0543$ has been chosen to minimize the error (the gradient correction is not as beneficial in 1D as in 3D). For T^{ML}, both the mean and maximum absolute errors improve as N or M increases (the system becomes more uniform as $N \to \infty$ [3]). At $M = 80$, we have already

TABLE I. Parameters and errors (mean absolute, standard deviation, and max absolute in kcal/mol) as a function of electron number N and number of training densities M. Brackets represent errors on self-consistent densities with $m = 30$ and $\ell = 5$. The α_j are on the order of 10^6 and both positive and negative [18].

N	M	$\lambda \times 10^{14}$	σ	$\overline{\lvert\Delta T\rvert}$	$\lvert\Delta T\rvert^{\text{std}}$	$\lvert\Delta T\rvert^{\text{max}}$
1	40	57 600	238	3.3	3.0	23
1	60	10 000	95	1.2	1.2	10
1	80	4489	48	0.43	0.54	7.1
1	100	12	43	0.15[3.0]	0.24[5.3]	3.2[46]
1	150	6.3	33	0.06	0.10	1.3
1	200	3.2	28	0.03	0.05	0.65
2	100	1.7	52	0.13[1.4]	0.20[3.0]	1.8[37]
3	100	4.0	74	0.12[0.9]	0.18[1.5]	1.8[14]
4	100	2.0	73	0.08[0.6]	0.14[0.8]	2.3[6]
1–4[a]	400	3.2	47	0.12	0.20	3.6

[a]Training set includes $\boldsymbol{n}_{j,N}$, for $j = 1, \ldots, 100$, $N = 1, \ldots, 4$.

achieved "chemical accuracy," i.e., a MAE below 1 kcal/mol. At $M = 200$, no error is above 1 kcal/mol. Simultaneously incorporating different N into the training set has little effect on the overall performance, and we stop at $N = 4$ merely for convenience. Note that our accuracy is so high that energy differences due to very subtle density changes are accurately captured by our approximation.

With such unheard of accuracy, it is tempting to declare "mission accomplished," but this would be premature. A KE functional that predicts only the energy is useless in practice, since orbital-free DFT uses functional derivatives in self-consistent procedures to find the density within a given approximation, via

$$\frac{\delta T[n]}{\delta n(x)} = \mu - v(x), \tag{6}$$

where μ is adjusted to produce the required particle number. The (discretized) functional derivative of T^{ML} is

$$\frac{1}{\Delta x}\nabla_{\boldsymbol{n}} T^{\text{ML}}(\boldsymbol{n}) = \sum_{j=1}^{M} \alpha'_j(\boldsymbol{n}_j - \boldsymbol{n})k(\boldsymbol{n}_j, \boldsymbol{n}), \tag{7}$$

where $\alpha'_j = \alpha_j/(\sigma^2 \Delta x)$. This oscillates wildly relative to the exact curve (Fig. 3), typical behavior that does not improve with increasing M. No finite interpolation can accurately reproduce all details of a functional derivative, and this behavior probably worsens when more varied densities are treated.

We overcome this problem using principal component analysis (PCA). The space of all densities is contained in $\mathbb{R}^G$, but only a few directions in this space are relevant. For a given density $\boldsymbol{n}$, find the m training densities $(\boldsymbol{n}_{j_1}, \ldots, \boldsymbol{n}_{j_m})$ closest to $\boldsymbol{n}$. Construct the covariance matrix of directions from $\boldsymbol{n}$ to each training density $\boldsymbol{C} = \boldsymbol{X}^\top \boldsymbol{X}/m$, where $\boldsymbol{X} = (\boldsymbol{n}_{j_1} - \boldsymbol{n}, \ldots, \boldsymbol{n}_{j_m} - \boldsymbol{n})^\top$. Diagonalizing $\boldsymbol{C} \in \mathbb{R}^{G\times G}$ gives eigenvalues λ_j and eigenvectors $\boldsymbol{x}_j$ that we list in decreasing order. The $\boldsymbol{x}_j$ with larger λ_j are directions with substantial variation in the data set. Those with λ_j below a cutoff are irrelevant [18]. In these extraneous dimensions, there is too little variation within the data set, producing noise in the model functional derivative. By projecting onto the subspace spanned by the relevant dimensions, we eliminate this noise. This projection is given by $\boldsymbol{P}_{m,\ell}(\boldsymbol{n}) = \boldsymbol{V}^\top \boldsymbol{V}$ where $\boldsymbol{V} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_\ell)^\top$ and ℓ is the number of relevant eigenvectors. In Fig 1, with $m = 30$ and $\ell = 5$, the projected functional derivatives are in excellent agreement.

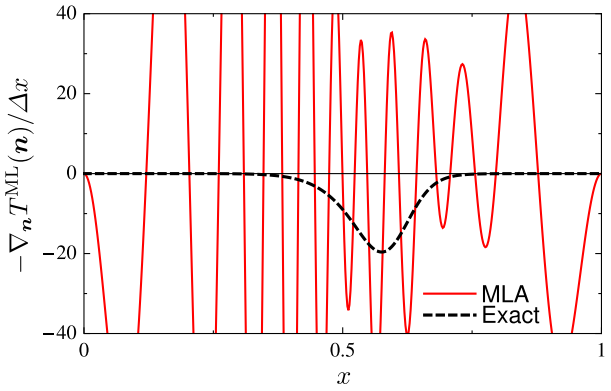

FIG. 3 (color online). Functional derivative of T^{ML}, evaluated on the density of Fig. 2.

The ultimate test for a density functional is the error of the functional evaluated on the self-consistent density that minimizes the total energy. This error will be several times larger than that of the functional evaluated on the exact density. For example, T^{loc} on particles in 1D flat boxes always gives a four-times larger error. To find a minimizing density, we perform a gradient descent search that is restricted to the local PCA subspace. Starting from a guess $\boldsymbol{n}^{(0)}$, take a small step in the opposite direction of the projected functional derivative of the total energy in each iteration j as follows:

$$\boldsymbol{n}^{(j+1)} = \boldsymbol{n}^{(j)} - \boldsymbol{\epsilon}\boldsymbol{P}_{m,\ell}(\boldsymbol{n}^{(j)})[\boldsymbol{v} + \nabla_{\boldsymbol{n}} T^{\text{ML}}(\boldsymbol{n}^{(j)})/\Delta x], \tag{8}$$

where $\boldsymbol{\epsilon}$ is a small number and $\boldsymbol{v}$ is the discretized potential. The search is unstable if ℓ is too large, inaccurate if ℓ is too small, and relatively insensitive to m [18].

The performance of T^{ML} in finding self-consistent densities is given in Table I. The errors are an order of magnitude larger than that of T^{ML} on the exact densities. We do not find a unique density, but instead a set of similar densities depending on the initial guess (e.g., Fig. 2). The density with the lowest total energy does not have the smallest error. Although the search does not produce a unique minimum, it produces a range of similar but valid approximate densities, each with a small error. Even with an order of magnitude larger error, we still reach chemical accuracy, now on self-consistent densities. No existing KE approximation comes close to this performance.

What are the limitations of this approach? ML is a balanced interpolation on known data, and it should be unreliable for densities far from the training set. To demonstrate this, we generate a new data set of 5000 densities with $N = 1$ for an expanded parameter range: $0.1 < a < 20$, $0.2 < b < 0.8$, and $0.01 < c < 0.3$. The predictive variance (borrowed from Gaussian process regression [20])

$$\mathbb{V}[T^{\mathrm{ML}}(\boldsymbol{n})] = k(\boldsymbol{n}, \boldsymbol{n}) - \boldsymbol{k}(\boldsymbol{n})^{\top}(\boldsymbol{K} + \lambda \boldsymbol{I})^{-1}\boldsymbol{k}(\boldsymbol{n}), \quad (9)$$

where $\boldsymbol{k}(\boldsymbol{n}) = (k(\boldsymbol{n}_1, \boldsymbol{n}), \ldots, k(\boldsymbol{n}_M, \boldsymbol{n}))$, is a measure of the uncertainty in the prediction $T^{\mathrm{ML}}(\boldsymbol{n})$ due to sparseness of training densities around $\boldsymbol{n}$. In Fig. 4, we plot the error ΔT as a function of $\log\{\mathbb{V}[T^{\mathrm{ML}}(\boldsymbol{n})]\}$, for both the test set and the new data set, showing a clear correlation. From the inset, we expect our MLA to deliver chemical accuracy for $\log\{\mathbb{V}[T^{\mathrm{ML}}(\boldsymbol{n})]\} < -24$.

Does ML allow for human intuition? In fact, the more prior knowledge we insert into the MLA, the higher the accuracy we can achieve. Writing $T = T^W + T_\theta$, where $T_\theta \geq 0$ [14], we repeat our calculations to find a MLA for T_θ. For $N = 1$, we get almost zero error and a factor of 2–4 reduction of error otherwise. Thus, intuition about the functional can be built in to improve results.

The primary interest in KS DFT is XC for molecules and solids. We have far less information about this than in the prototype studied here. For small molecules and simple solids, direct solutions of the Schrödinger equation yield highly accurate values of E_{XC}. Imagine a sequence of models, beginning with atoms, diatomics, etc., in which such accurate results are used as training data for a MLA. In the case of XC, the key issues are how accurate a functional can be attained with a finite number of data, and what fraction of the density space it is accurate for.

A more immediate target is the noninteracting KE in KS DFT calculations. An accurate approximation would allow finding densities and energies without solving the KS equations, greatly increasing the speed of large calculations [14]. The key differences with our prototype is the three-dimensional nature, the Coulomb singularities, and the variation with nuclear positions. For this problem, finding self-consistent densities is crucial, and hence our focus here. But in the 3D case, every KS calculation ever run, including every iteration in a self-consistent loop, generates training data—a density, KE, KS potential, and functional derivative. The space of all systems of practical interest, including both solids and molecules, is vast, but can be approached in small steps, including always training on "nearby" densities.

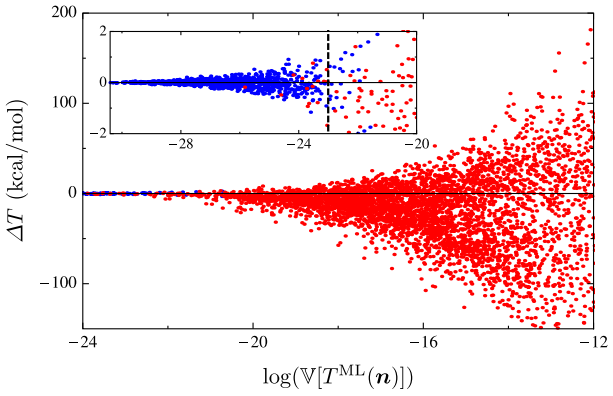

FIG. 4 (color online). The correlation between MLA error and predictive variance for $N = 1$, $M = 100$. Each point represents a density in the test set (blue) or new data set (red). The vertical line denotes the transition between interpolation and extrapolation.

Continuing the discussion of the KE functional, our demo has been (purposely) limited to a very simple class of potentials. But unlike traditional fitting to limited approximate forms of a functional, there is no reason *a priori* to expect our method to scale poorly with the complexity of the one-body potential. In ML, the problem is reduced to approximating a functional by a scalar function of a high-dimensional domain (500 here). The difficulty depends on how smooth this functional is, which determines how many training densities we need to interpolate accurately. We estimate the effective dimensionality, or RDE [21], of our data at about 12. We anticipate this to increase by a modest factor when dealing with electrons of differing character (e.g., d and f electrons), but not exponentially, for the weakly correlated systems for which present XC functionals are useful. Moreover, statistical learning theory shows [17,22] that the error of regression estimators (i.e., our method) scales asymptotically as $1/M$ with the number of training data M for faithful models and as $1/\sqrt{M}$ for unfaithful ones. As is customary in ML, none of these questions will be answered until the full problem has been attempted. Preliminary model calculations for bond dissociation, where most present approximations fail due to their local nature, show only a mild increase in the need for training data [23].

Two last points. The first is that this type of empiricism is qualitatively distinct from that present in the literature [10]. The choices we made are those customary in ML and require no intuition about the physical nature of the problem. Second, the approximation is expressed in terms of about 10^5 numbers, and only the projected functional derivative is accurate. We have no simple way of comparing such approximations to those presently popular. For example, for $N = 1$ in the prototype, the exact functional is T^W. How is this related to our MLA, and how does our MLA account for this exact limit?

The authors thank the Institute for Pure and Applied Mathematics at UCLA for hospitality and acknowledge NSF Grant No. CHE-1112442 and NRF Korea Grant No. 2010-220-C00017 (J. S. and K. B.), EU PASCAL2 and DFG Grant No. MU 987/4-2 (M. R., K. H., and K. R. M.), EU Marie Curie Grant No. IEF 273039 (M. R.), and NRF Korea Grant No. R31-10008 (K. R. M.).

[1] P. Hohenberg and W. Kohn, Phys. Rev. **136**, B864 (1964).
[2] W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965).
[3] P. Elliott, D. Lee, A. Cangi, and K. Burke, Phys. Rev. Lett. **100**, 256406 (2008).
[4] P. J. Stephens, F. J. Devlin, C. F. Chabalowski, and M. J. Frisch, J. Phys. Chem. **98**, 11623 (1994).
[5] J. P. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996).
[6] J. P. Perdew and A. Ruzsinszky, Int. J. Quantum Chem. **110**, 2801 (2010).
[7] A. D. Becke, Phys. Rev. A **38**, 3098 (1988).
[8] C. Lee, W. Yang, and R. G. Parr, Phys. Rev. B **37**, 785 (1988).
[9] Y. Zhao and D. Truhlar, Theor. Chem. Accounts **120**, 215 (2007).
[10] K. Burke, J. Chem. Phys. **136**, 150901 (2012).
[11] K. -R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, IEEE Trans. Neural Network **12**, 181 (2001).
[12] O. Ivanciuc, in *Reviews in Computational Chemistry*, edited by K. Lipkowitz and T. Cundari (John Wiley & Sons, Hoboken, 2007), Vol. 23, p. 291.
[13] M. Rupp, A. Tkatchenko, K. -R. Müller, and O. A. von Lilienfeld, Phys. Rev. Lett. **108**, 058301 (2012).
[14] V. Karasiev, R. Jones, S. Trickey, and F. Harris, in *New Developments in Quantum Chemistry*, edited by J. Paz and A. Hernández (Research Signpost, Kerala, to be published).
[15] R. M. Dreizler and E. K. U. Gross, *Density Functional Theory: An Approach to the Quantum Many-Body Problem* (Springer, Berlin, 1990).
[16] E. Hairer, P. Nørsett, P. Syvert Paul, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems* (Springer, New York, 1993).
[17] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning. Data Mining, Inference, and Prediction* (Springer, New York, 2009), 2nd ed..
[18] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevLett.108.253002 for information necessary to construct the MLA functional and more detail on the PCA projections and self-consistent densities.
[19] D. Lee, L. A. Constantin, J. P. Perdew, and K. Burke, J. Chem. Phys. **130**, 034107 (2009).
[20] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning* (MIT Press, Cambridge, 2006).
[21] M. Braun, J. Buhmann, and K. -R. Müller, J. Mach. Learn. Res. **9**, 1875 (2008).
[22] K. -R. Müller, N. Murata, M. Finke, K. Schulten, and S. Amari, Neural Comput. **8**, 1085 (1996).
[23] J. C. Snyder, M. Rupp, K. Hansen, K. -R. Müller, and K. Burke (to be published).