Understanding Machine-learned Density Functionals

Li Li,[1] John C. Snyder,[1,2] Isabelle M. Pelaschier,[3,1] Jessica Huang,[2] Uma-Naresh Niranjan,[4] Paul Duncan,[2] Matthias Rupp,[5] Klaus-Robert Müller,[6,7] and Kieron Burke[2,1]

[1]*Department of Physics and Astronomy, University of California, Irvine, CA 92697*
[2]*Department of Chemistry, University of California, Irvine, CA 92697*
[3]*Department of Physics, Vanderbilt University, Nashville, TN 37235, USA*
[4]*Department of Computer Science, University of California, Irvine, CA 92697*
[5]*Department of Chemistry, University of Basel, Klingelbergstr. 80, 4056 Basel, Switzerland*
[6]*Machine Learning Group, Technical University of Berlin, 10587 Berlin, Germany*
[7]*Department of Brain and Cognitive Engineering, Korea University, Anam-dong, Seongbuk-gu, Seoul 136-713, Korea*

(Dated: April 2, 2014)

Kernel ridge regression is used to approximate the kinetic energy of non-interacting fermions in a one-dimensional box as a functional of their density. The properties of different kernels and methods of cross-validation are explored, and highly accurate energies are achieved. Accurate *constrained optimal densities* are found via a modified Euler-Lagrange constrained minimization of the total energy. A projected gradient descent algorithm is derived using local principal component analysis. Additionally, a sparse grid representation of the density can be used without degrading the performance of the methods. The implications for machine-learned density functional approximations are discussed.

PACS numbers: 31.15.E-, 31.15.X-, 02.60.Gf, 89.20.Ff

CONTENTS

I. INTRODUCTION

Since the early days of quantum mechanics, it has been known that sufficiently accurate solutions of Schrödinger's equation for electrons and nuclei yield good predictions of the properties of solids and molecules [1]. But the Coulomb repulsion between electrons causes the computational cost of solving the Schrödinger equation to grow rapidly with the number of electrons, N [2]. However, as Hohenberg and Kohn proved in 1964 [3], the one-electron density may be used as the basic variable of quantum mechanics instead of the wavefunction, greatly reducing the complexity of the computational problem. This is called density functional theory (DFT) [4]. In principle, the mapping of the Schrödinger equation to one with the electron density is exact, but in practice, both the kinetic energy and the energy of the interaction between electrons must be approximated. In the original Thomas-Fermi theory [5, 6], a local density functional approximation to the kinetic energy is used. However, Thomas-Fermi theory proved unsuitable for chemical and solid-state applications as it does not bind matter [7]. Shortly after the Hohenberg-Kohn theorems, Kohn and Sham (KS) [8] found a middle ground by mapping the many-body system onto a fictitious system of non-interacting electrons which reproduce the exact electron density. The main reason KS DFT became successful is because the kinetic energy of these non-interacting electrons is an excellent approximation to the many-body kinetic energy. Simple approximations to the interaction energy produce much greater accuracy and reliability compared with the standard orbital-free DFT schemes built on Thomas-Fermi theory. However, the accuracy of the results are still sensitive to the approximation of the exchange-correlation (XC) functional. In the past four decades, there has been extensive research into improving density functional XC approximations. Development of both empirical and non-empirical functionals require

great intuition built on years of experience, as well as painstaking trial and error [9–11].

Despite the great success KS DFT has enjoyed, the computational cost scales as $O(N^3)$, which is much worse than the linear scaling of orbital-free DFT [12]. Thus, there continues to be strong interest in improving upon existing orbital-free approximations to the kinetic energy [12–14]. A sufficiently accurate approximation to $T_{\rm S}[n]$, the kinetic energy of KS electrons as a functional of the ground-state density $n(\mathbf{r})$ would enable highly accurate orbital-free DFT calculations with the same accuracy as KS DFT at a fraction of the computational cost. For example, benchmark orbital-free DFT calculations are capable of treating millions of atoms in metals [15] or proteins in solvent [16]. Note that accuracy in $T_{\rm S}$ beyond that of current XC approximations would be unnecessary, since all standard orbital-free DFT schemes utilize the KS decomposition of the energy, so that standard XC approximations developed for KS DFT can be utilized. However, since $T_{\rm S}$ is typically comparable to the total energy of the system [4], an unavoidable problem is that a useful kinetic energy (KE) functional calls for much stricter relative accuracy than XC functionals. Additionally, accurate functional derivatives are required because one finds the ground state density by solving an Euler equation with the approximate kinetic energy functional. Continued efforts have been made in this research direction, with some notable progress [17–28]. For a review of state-of-the-art orbital-free DFT functionals, we refer the reader to Ref. [12].

In DFT, functionals typically fall into two categories. Non-empirical functionals derived from first principles tend to work well across a broad range of systems, and may exhibit systemic errors in treating certain types of interactions. Semi-empirical functionals introduce parameters that are fitted to standard data sets, and are typically more accurate with less systematic errors.

Recently, some of us applied machine learning (ML) in a completely new approach to approximating density functionals [29, 30]. In a proof of principle, kernel ridge regression was used to approximate the kinetic energy of non-interacting fermions confined to a 1d box as a functional of the electron density [29]. In that work, a modified orbital-free DFT scheme was able to produce highly accurate self-consistent densities and energies that were systematically improvable with additional training data. ML algorithms are capable of learning high-dimensional patterns by non-linear interpolation between given data. These powerful methods have proved to be very successful in many applications [31], including medical diagnoses [32], stock market predictions [33], automated text categorization [34], and others. Recently, ML has been applied to quantum chemistry, including fast and accurate modeling of molecular atomization energies [35–37], optimizing transition state theory dividing surfaces [38], and calculating bulk crystal properties at high temperatures [39].

This new approach to density functional approximation suffers none of the typical challenges found in traditional approximations, but presents many new ones. First and foremost, ML is data-driven: reference calculations are needed to build a model for the KE functional. Since every iteration in a KS DFT calculation provides an electron density and its exact non-interacting KE, reference data is relatively easy to obtain. Additionally, the ML approximation (MLA) to the KE may have thousands or millions of parameters and satisfy none of the standard exact conditions in DFT, such as positivity, scaling, and exactness for a uniform electron gas. On the other hand, the form of the MLA is completely general and thus directly approximates the functional itself, suffering none of the typical issues plagued by standard functionals starting from a local approximation. For example, some of us recently showed that an MLA for the KE has no problem accurately dissociating soft-Coulomb diatomics in 1d—a huge challenge for standard approximations [30]. However, kernel ridge regression is strictly a method of interpolation. An MLA can only be used on systems it was designed for.

In this paper, we explore the properties of the MLA derived in Ref. [29] in greater detail. In particular, we investigate the use of various kernels and their properties and the efficiency of various cross validation methods. We discuss the issue of functional derivatives of the MLA in greater detail, and explain how a modified constraint to the standard Euler equation enables highly accurate self-consistent densities, or *constrained optimal densities*, to be found. Additionally, a projected gradient descent algorithm is derived using local principal component analysis in order to solve the modified Euler equation. Finally, we explore the use of a sparse grid representation of the electron density and its effects on the method.

II. THEORY AND BACKGROUND

Throughout this work, we consider only non-interacting same-spin fermions in one-dimension. Thus, all electron densities $n(x)$ are fully spin-polarized. Atomic units are used in symbolic equations, but energies are usually presented in kcal/mol.

A. Model system

Consider N non-interacting same-spin fermions subject to a smooth external potential in one-dimension, with hard walls at $x = 0$ and $x = 1$. We restrict this study to a simple class of potentials, namely a sum of 3 Gaussian dips with varying heights, widths and centers:

$$v(x) = \sum_{j=1}^{3} a_j \exp(-(x - b_j)^2/(2c_j^2)), \tag{1}$$

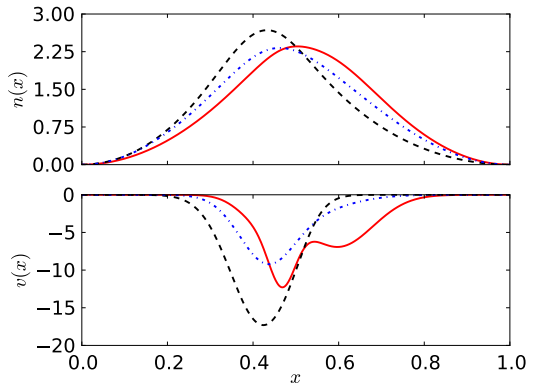

FIG. 1. A few sample densities and their corresponding potentials, for $N = 1$.

for $x \in [0,1]$, and $v(x) = \infty$ elsewhere. The Hamiltonian for this system is simply $\hat{H} = \hat{T} + \hat{V}$, where $\hat{T} = -\partial^2/2\partial x^2$ and $\hat{V} = v(x)$. We solve the Schrödinger equation

$$\left(-\frac{1}{2}\frac{\partial^2}{\partial x^2} + v(x)\right)\phi(x) = \epsilon\phi(x), \tag{2}$$

for the eigenvalues ϵ_j and orbitals $\phi_j(x)$. As our fermions are same-spin, each orbital $\phi_j(x)$ is singly-occupied. Thus, the electron density is given by

$$n(x) = \sum_{j=1}^{N} |\phi_j(x)|^2, \tag{3}$$

and the kinetic energy is

$$T = \frac{1}{2}\sum_{j=1}^{N}\int_0^1 dx|\phi_j'(x)|^2. \tag{4}$$

A dataset is created by randomly sampling $a_j \in [1,10], b_j \in [0.4,0.6], c_j \in [0.03,0.1]$, to generate 2000 different potentials. For each potential, the system is occupied with up to 4 fermions, and the exact densities and kinetic energies are computed. Numerically, the Schrödinger equation is solved by discretizing the density on a grid:

$$x_j = (j-1)/(N_G-1), \quad j = 1,\ldots,N_G \tag{5}$$

and $\Delta x = 1/(N_G-1)$ is the grid spacing. Numerov's method [40] together with a shooting method is used to solve for the eigenvalues and eigenfunctions of Eq. (2). $N_G = 500$, the error in our reference kinetic energies is less than 10^{-7}. Fig. 1 gives a few sample densities and their corresponding potentials.

The data used here is identical to that of Ref. [29]. The exact values of the parameters used in each sample is given in the supplementary information of Ref. [29]. Of the 2000 samples generated, the first half is reserved for training while the second half is reserved for testing (which we refer to as the *test set*).

B. Orbital-free DFT

In orbital-free DFT, T_s is approximated as a functional of $n(x)$. For our model system with non-interacting fermions, the total energy is given as

$$E_v = \min_n \{T[n] + V[n]\}, \tag{6}$$

for a given potential $v(x)$. The potential is known exactly as a functional of $n(x)$:

$$V[n] = \int_0^1 dx\, n(x)v(x). \tag{7}$$

Via the variational principle, the ground-state density is found by the Euler-Lagrange constrained search

$$\delta\left\{E_v[n] - \mu\left(\int n(x)\,dx - N\right)\right\} = 0, \tag{8}$$

where the chemical potential μ is adjusted to produce the required particle number N. This becomes simply

$$\frac{\delta T[n]}{\delta n(x)} = \mu - v(x). \tag{9}$$

The density that satisfies this equation, minimizing $E_v[n]$ with the normalization constraint, is found self consistently.

Given the exact functional $T[n]$, solving Eq. (9) will yield the exact ground-state density of the system. But in practice, T must be approximated. Let $\tilde{T}$ be such an approximation, $n(x)$ be the exact density, and $\tilde{n}(x)$ be the self-consistent density found with $\tilde{T}$. There are two measures of the error of such an approximate $\tilde{T}$ [41]. The first is to compute the functional-driven error $\Delta T_F = \tilde{T}[n] - T[n]$, which is simply the error in the KE evaluated on the exact density. The second (and much more difficult) test is to insert $\tilde{T}$ into Eq. (9), solve for the approximate density $\tilde{n}$, and compute its error relative to the KE of the exact density $\Delta E = \tilde{E}_v[\tilde{n}] - E_v[n]$. Then the density-driven error is defined as $\Delta E_D = \Delta E - \Delta T_F$ [41]. This is the additional error incurred by the approximate density. In practice, a functional which only satisfies the first test is not much use, as the ground-state density itself must also be obtained from this approximation. In orbital-free DFT, self-consistent results can be much worse than energies of KS densities, as inaccuracies in the functional derivative can cause large errors in the corresponding density. In the case of the KE functional for real systems, functional derivatives of traditional approximations can have singularities at the nuclei, making all-electron calculations very difficult, if not impossible, to converge [12]. Many of these problems can be avoided through use of pseudopotentials [12, 28], but in general the solution for Eq. (9) is nontrivial.

As mentioned above, the simplest density functional approximation to T_s is the local approximation [4], which

for spin-polarized densities in 1d is

$$T^{\rm loc}[n] = \frac{\pi^2}{6}\int dx\, n^3(x). \tag{10}$$

For $N = 1$, the exact KE has the von Weisäcker [17] form:

$$T^{\rm W}[n] = \int dx\, \frac{n'(x)^2}{8n(x)}. \tag{11}$$

As was shown in Ref. [29], the local approximation does poorly. The mean absolute error (MAE) on the test set is 217 kcal/mol, and self-consistent results are even worse at 1903 kcal/mol. A standard extension of the local approximation to a semi-local form is to add a fraction of $T^{\rm W}[n]$ to $T^{\rm loc}[n]$, forming a modified gradient expansion approximation. It was shown in Ref. [29] that this did little to improve upon the local approximation.

C. Data topology and representation

Typically in ML, the data has a finite representation. For example, in Ref. [35], molecular structures are represented by a Coulomb matrix and the model predicts atomization energies. In contrast, the electronic density $n(x)$ is a continuous function restricted to the domain [42]

$$\mathcal{J}_N \equiv \left\{ n \,\middle|\, n(x) \geq 0, n^{1/2}(x) \in H^1(\mathbb{R}), \int n(x)\,dx = N \right\}, \tag{12}$$

where $H^1(\mathbb{R})$ is a Sobolev space[1]. Although $\mathcal{J}_N$ is infinite dimensional, in practice $n(x)$ is expanded in a finite basis (with N_G basis functions). In this work, we use a real space grid to represent the density, since our reference calculations are done using the same grid. We use the L^2 inner product and norm between densities $n_i(x), n_j(x)$

$$\langle n_i, n_j\rangle = \int_{-\infty}^{\infty} dx\, n_i(x) n_j(x), \quad \|n\| = \sqrt{\langle n, n\rangle}. \tag{13}$$

(In actual calculations, all densities are represented on a finite basis, and thus will have have a finite L^2-norm). Since the ML algorithm is expressed in terms of this inner product, the results are independent of the specific representation used as long as the basis is converged.

Even with a truncated basis, $\mathcal{J}_N$ is still high-dimensional and applying ML to learn the KE of all densities in $\mathcal{J}_N$ would not be feasible. Fortunately, we are only interested in a subspace of $\mathcal{J}_N$ related to a specific class of potentials (e.g. Gaussian dips), which greatly reduces the variety of possible densities. In general, let the potential $v(x)$ be parametrized by the parameters $\{p_1, \ldots, p_d\}$. We define the density manifold $\mathcal{M}_N \subset \mathcal{J}_N$ as the set of all densities that come from these potentials with a given particle number N. In general, $\mathcal{M}_N$ is a d-dimensional manifold. The training densities, $n_j(x)$ for $j = 1, \ldots, N_T$, are sampled from $\mathcal{M}_N$. In the present work, the external potential has 9 parameters, and thus d is at most 9.

[1] A Sobolev space $W^{k,p}(\mathbb{R})$ is a vector space of functions with a norm that is a combination of L^p-norms of the function itself and its derivatives up to a given order k. It is conventional to write $W^{1,2}(\mathbb{R})$ as $H^1(\mathbb{R})$. $f \in H^1(\mathbb{R})$ means that f and its first order derivative are in L^2.

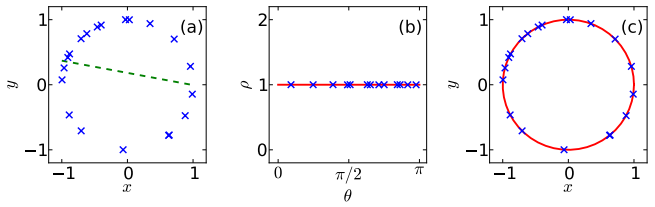

FIG. 2. Example of the non-linear transformation of data to feature space. (a) The data is non-linear (a circle) in Cartesian coordinates. The green dashed line is a linear fit to the data points (blue crosses). (b) When the data is transformed to feature space by $x \to \rho\cos\theta$, $y \to \rho\sin\theta$, the linear structure in the data is revealed (red solid line). (c) The model can be transformed back to the original space to give a non-linear fit of the data.

D. The kernel trick and feature space

In finding the structure of low-dimensional data, it is often sufficient to optimize parametrized non-linear forms (e.g., using a polynomial to fit a sinusoid). For high-dimensional, nonlinear data this becomes increasingly difficult. In machine learning, the approach is to transform the *data itself* non-linearly to a high-dimensional space known as feature space, such that the data becomes linear [31, 43–45].

Fig. 2 illustrates data points that lie on a circle in the Cartesian plane. As shown, the data becomes linear on transformation to polar coordinates, and linear regression can subsequently be used to fit the data. Transforming back to Cartesian coordinates recovers the non-linearity. Let the data points belong to a vector space χ, also called input space, and let $\Phi : \chi \to F$ be the map to feature space F. Assuming we wish to apply a linear method such as regression in feature space F, we note that regression can be expressed solely in terms of the inner product between feature vectors $\Phi(x)$ and $\Phi(y)$, where $x, y \in \chi$. We define the kernel k such that

$$k(x, y) = \langle \Phi(x), \Phi(y)\rangle. \tag{14}$$

The kernel can generally be thought of a measure of similarity between data, but must satisfy Mercer's condition:

$$\int\int k(x, y) g(x) g(y) dx dy \geq 0, \tag{15}$$

for all $g(x)$ satisfying $\int_{-\infty}^{\infty}|g(x)|^2dx<\infty$. Mercer's theorem [46] guarantees the existence of a feature space F, which is a reproducing kernel Hilbert space [47]. Since the linear algorithm in F may be expressed in terms of the kernel in Eq. (14), Φ need never be explicitly computed. This procedure is known as the *kernel trick*, and enables easy nonlinearization of all linear scalar product-based methods that can be expressed via an inner product [48].

E. Kernel ridge regression

Kernel ridge regression is the nonlinear version of regression with a regularization term to prevent overfitting [49]. Our MLA for the KE has the form

$$T^{\mathrm{ML}}[n]=\sum_{j=1}^{N_T}\alpha_j k[n,n_j], \tag{16}$$

where N_T is the number of training densities, α_j are weights to be determined, n_j are training densities and $k[n,n_j]$ is the kernel. The weights are found by minimizing the quadratic cost plus regularization

$$\mathcal{C}(\boldsymbol{\alpha})=\sum_{j=1}^{M}(T^{\mathrm{ML}}[n_j]-T[n_j])^2+\lambda\boldsymbol{\alpha}^\top\boldsymbol{K}\boldsymbol{\alpha}, \tag{17}$$

where $\boldsymbol{\alpha}=(\alpha_1,\ldots,\alpha_{N_T})$ and $\boldsymbol{K}$ is the kernel matrix, $\boldsymbol{K}_{ij}=k[n_i,n_j]$, and λ is called the regularization strength. The second term penalizes weights with large magnitudes in order to prevent overfitting.[2] Minimizing $\mathcal{C}(\boldsymbol{\alpha})$ gives

$$\boldsymbol{\alpha}=(\boldsymbol{K}+\lambda\boldsymbol{I})^{-1}\boldsymbol{T}, \tag{18}$$

where $\boldsymbol{I}$ is the identity matrix and $\boldsymbol{T}=(T[n_1],\ldots,T[n_{N_T}])$. The *hyperparameters*, which includes the regularization strength λ and the parameters of the kernel such as the length scale σ, are found via cross validation (see [36] and Sect. III).

The choice of the kernel will depend on the given data. Some kernels are designed to be generally robust and applicable (e.g., the Gaussian kernel), while others are designed for a specific type of data (see e.g. [31, 51]). A good choice of kernel can reflect the characteristics of the data (see [52]). In Ref. [29], we chose the Gaussian kernel

$$k[n_i,n_j]=\exp\left(-\|n_i-n_j\|^2/2\sigma^2\right), \tag{19}$$

[2] The regularization term accounts for the possibility of noisy data (e.g. experimental data), and imposes certain smoothness conditions on the model (see [50]). Our reference data is deterministic and thus noise-free in this sense, but, because the precision of our calculations is limited, we may consider the numerical uncertainty to be noise.

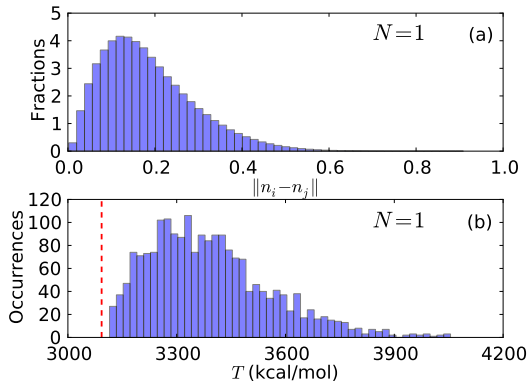

FIG. 3. (a) Normalized distribution of the Euclidean distance between all distinct pair of densities in the dataset (2000 densities). The maximum distance between any pair is 0.9. (b) Histogram of the KE in the dataset. The vertical dashed line at 3093 kcal/mol is the ground-state energy of one fermion in a flat box of length 1.

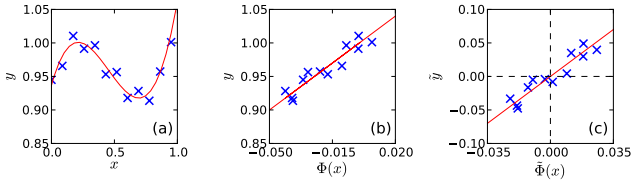

FIG. 4. (a) An example 1d noisy data set. (b) Transformation to feature space $\Phi(x)$. (c) Centering of data in feature space.

where σ is the length scale. Since the density is represented on a uniform grid, the L^2-norm can be approximated by[3]

$$\|n_i-n_j\|^2=\Delta x\sum_{l=1}^{N_G}(n_i(x_l)-n_j(x_l))^2 \tag{20}$$

where x_l is given by the grid defined in Eq. (5). This approximation becomes exact as $\Delta x\to 0$. Fig. 3 shows the range and distribution of Euclidean distances between all pairs of densities and KE of all densities in the dataset with $N=1$.

Ordinary linear regression models frequently employ a bias term to account for the fact that the data might lie away from the origin. Without this term, the regression line is forced to go through the origin, causing a

[3] Note that, in Ref. [29], the same representation for the density was used, but the density were treated as vectors, so the standard Euclidean distance was used in the kernel. This is equivalent to the formulation here, except our notation is more general now (e.g. Simpson's rule could be used to approximation the L^2-norm instead of a Riemann sum), and the length scale in Gaussian kernel here is related to the scale of the kernel in Ref. [29] by a factor of $\sqrt{\Delta x}$.

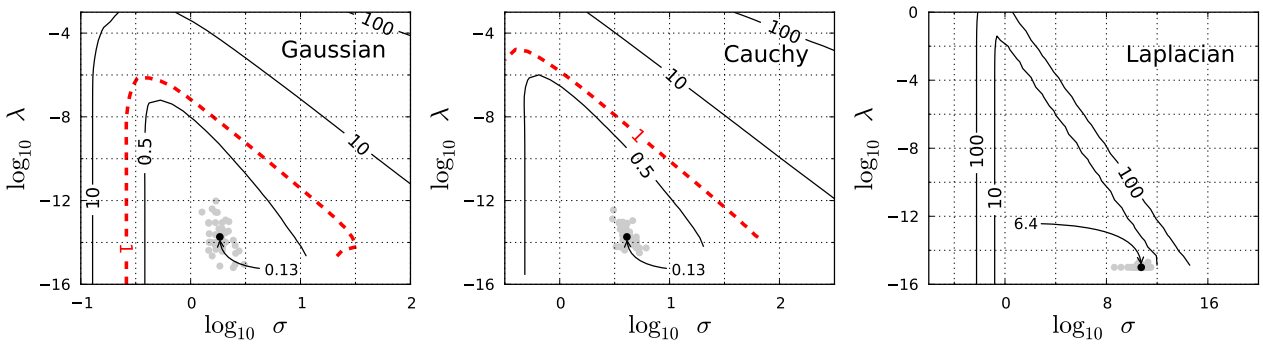

FIG. 5. Contour plots of the functional-driven MAE $\overline{|\Delta T_F|}$ over the test set in kcal/mol for selected kernels with $N_T = 100$. The dashed line delineates the region where the model achieves chemical accuracy. Each gray dot gives the optimal choice of hyperparameters from a randomized 10-fold cross validation. The black dot denotes the median over 40 repetitions. In the lower right region (i.e. small λ and large σ), the matrix inverse in Eq. (18) is numerically unstable due to the limited precision of the calculation.

systematic error if the data does not. The bias term can be implemented directly, or by centering the samples such that the mean is zero. Fig. 4 illustrates the transformation to feature space for an example 1d data set and linear regression in feature space. If the data is centered in feature space, the bias term is unnecessary. Here, we center the densities in features space such that $\sum_{j=1}^{N_T} \Phi(n_j) = 0$. We define the centered map to feature space $\tilde{\Phi}(n) = \Phi(n) - \sum_{j=1}^{N_T} \Phi(n_j)/N_T$. Then the centered kernel is [48]

$$\begin{aligned}\tilde{k}[n,n'] &= \langle \tilde{\Phi}(n), \tilde{\Phi}(n') \rangle \\ &= k[n,n'] - \frac{1}{N_T}\sum_{j}^{N_T}(k[n',n_j] + k[n,n_j]) \\ &\quad + \frac{1}{N_T^2}\sum_{i,j=1}^{N_T} k[n_i,n_j]. \end{aligned} \tag{21}$$

For simplicity, all equations given in this work assume that the data is centered (i.e. $k = \tilde{k}$). In fact, kernels such as the Gaussian kernel Eq.(19) whose induced reproducing kernel Hilbert space on a bounded domain is dense in the space of continuous functions on this domain do not require centering [53].

III. MODEL SELECTION

A. Kernels

Model selection refers to the process of selecting a kernel and the corresponding hyperparameters. In kernel ridge regression, this includes the regularization strength λ and the kernel parameters (e.g. in the Gaussian kernel, the length scale σ). Table I lists some standard kernels. Radial basis function (RBF) kernels, which include the Gaussian, Cauchy, and Laplacian kernels, all behave similarly and tend to work for a broad range of problems. Other kernels work well for specific data structures [31, 51] and regularization properties [48].

Kernel	$k[n,n']$
Gaussian	$\exp(-\|n-n'\|^2/2\sigma^2)$
Cauchy	$(1+\|n-n'\|^2/\sigma^2)^{-1}$
Laplacian	$\exp(-\|n-n'\|/2\sigma)$
Wave	$\frac{\theta}{\|n-n'\|}\sin\frac{\|n-n'\|}{\theta}$
Power	$\|n-n'\|^d$
Linear	$\langle n,n'\rangle$

TABLE I. Standard kernels. The parameters σ, θ, d are kernel parameters. The linear kernel has no parameters.

Fig. 5 shows the contours of the functional-driven MAE over the test set as a function of the regularization strength λ and the kernel parameter σ. We see that the qualitative behavior is similar for the Gaussian, Cauchy and Laplacian kernels. In the left region (where the contour lines are vertical), the length scale σ is much smaller than the distance between neighboring training densities. Thus the RBF-type kernel functions centered at each training density have minimal overlap, yielding a poor approximation to the KE functional. The kernel matrix becomes nearly unity, and the regularization λ has negligible effect. On the right side of the contour plot, the length scale is comparable to the global scale of the data. In these regions, the kernel functions are slowly varying and do not have enough flexibility to fit the nonlinearity in the data. The region with minimum MAE lies in the middle. The Gaussian and Cauchy kernels both give the same performance, with errors less than 1 kcal/mol in the middle region (enclosed by the dashed line), while the Laplacian kernel behaves poorly in comparison. This is likely due to the cusp in the form of the kernel, which cannot fit the smooth KE functional.

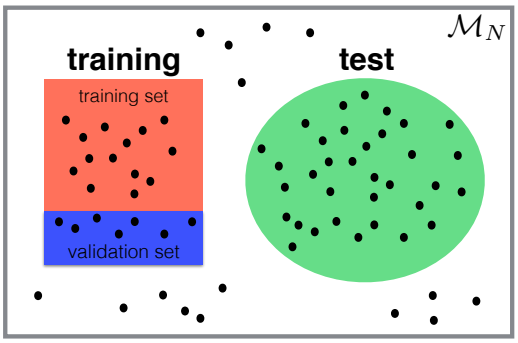

FIG. 6. Cartoon shows the relation of each data set in $\mathcal{M}_N$. Each black dot represents a sample (density and its corresponding KE). Training, validation and test set are subsets of the full data set.

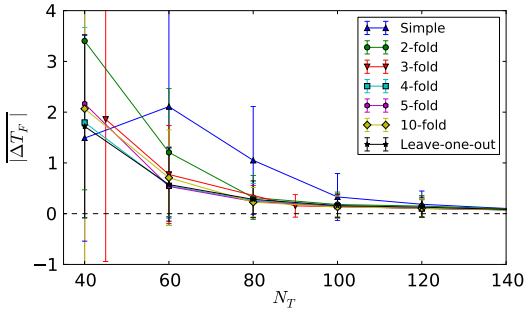

FIG. 7. Comparison of different cross validation methods, including simple, 2-fold, 3-fold, 4-fold, 5-fold, 10-fold and leave-one-out. The mean of the absolute functional-driven error $|\Delta T_F|$ (in kcal/mol) is evaluated on the test set and the error bars represent the standard deviation.

B. Optimization of hyperparameters

After picking a kernel family, the values of the hyperparameters must be chosen. Ideally, we select the hyperparameters such that the generalization error, which is the error not only on our training set but also on all future data, is minimal. The out-of-sample error must be estimated without looking at the test set (the test set is never touched during model selection, so that it can give a true test of the final performance of the model) [31, 36]. This procedure, known as cross-validation, is essential for model selection in preventing overoptimistic performance estimates (overfitting).

Various schemes for cross validation exist [31, 36, 54], but all obey a basic principle: the available data is subdivided into three parts: the *training*, *validation* and the *test sets*. The ML model is built from the training set and the hyperparameters are optimized by minimizing the error on the validation set (Fig. 6). The test set is never touched until the weights and hyperparameters have been determined. Then and only then, the generalization ability of the model can be assessed with the test data ([31, 36], see also [55]). Typically, the data is shuffled to ensure its random distribution between training and validation division. This can be repeated with different subdivisions. A few schemes, which will be analyzed for our kinetic energy functional estimation problem, are described below. For each scheme, a test set of 1000 samples is used to estimate the generalization error after the ML model is selected.

Simple cross validation: The training data (N_T samples) is randomly divided into a training set of 70% and a validation set of 30%. The hyperparameters are optimized by minimizing the MAE on the validation set.

k-fold cross validation:

Step 1: The N_T training data is randomly divided into k bins.

Step 2: The jth bin is used as the validation set and the remaining $k-1$ bins as training set. The model is built on the training set and the hyperparameters are selected by minimizing the MAE on the validation set.

Step 3: Repeat step 2 k times such that all bins have been used as validation sets. We will then have k models in total and the final hyperparameters are selected as the median over all models.

Because the mean cross validation error still depends on the initial random partitioning of data in cross validation, we repeat the procedure with different subdivisions [31].

Leave-one-out: Leave-one-out (LOO) is a special case of k-fold cross validation, when $k = N_T$. Thus each bin contains only one sample.

Typically, it is better to leave out as little data as possible to exploit the statistical power in the data. Simple cross validation is computationally expedient, but wasteful since not all training data participates in the optimization. k-fold cross validations are used in situations where data is very limited, or expensive to collect. Leave-one-out is often used with limited data and it becomes computationally intensive if N_T is large. k-fold cross validation gives a good balance on all counts.

For the Gaussian kernel, Fig. 7 shows the MAE on the test set with the hyperparameters optimized from different cross validation method. With 120 training densities, all schemes give a similar MAE, despite the large variations in σ and λ. This means that multiple models exist that give comparable performance. As expected, the only variations in MAE occur for more limited data.

Fig. 5 shows how 10-fold cross validation performs in selecting hyperparameters that generalize well to the test set, for a few kernels. The gray dots represent the optimal parameter choice for each repetition, and the black dot is the median over all repetitions. In this case, the global

Kernel	λ	p	$\overline{\lvert\Delta T_F\rvert}$	$\lvert\Delta T_F\rvert^{\max}$
Gaussian	$4.5 \cdot 10^{-14}$	1.6	0.13	3.4
Cauchy	$7.8 \cdot 10^{-14}$	3.5	0.13	2.9
Laplacian	$1.0 \cdot 10^{-15}$	$3.6 \cdot 10^{5}$	6.4	231
Linear	$6.2 \cdot 10^{-1}$	-	53.1	380
Wave	$4.5 \cdot 10^{-1}$	0.14	19.2	252
Power	$1.0 \cdot 10^{-13}$	1.96	3.3	104

TABLE II. The optimal hyperparameters found through 10-fold cross validation and the MAE over the test set for various kernels with $N = 1$ and $N_T = 100$. The kernel parameter p refers to σ for the Gaussian, Cauchy and Laplacian kernels, θ for the wave kernel and d for the power kernel. The linear kernel has no parameter. Errors are given in kcal/mol.

minimum of the MAE lies in a relatively flat basin. Each randomized cross validation lies near the true minimum, indicating the model generalizes well to the test set.

Finally, we use 10-fold cross validation (repeated 40 times) to optimize the hyperparameters. Table II shows the optimal hyperparameters and functional driven errors for the kernels listed in Table I. Some optimum values for the Gaussian kernel are listed in Table III. Detailed information with optimum values of other kernels are shown in supplementary material.

IV. RESULTS AND DISCUSSION

In the main work of this paper, we test in greater detail some of the methods that were introduced in Ref. [29] using only the Gaussian kernel, as it performs the best.

A. Errors on exact densities

In Table III, we evaluate our MLA, constructed using the first N_T training densities in our data set, on the exact densities of the test set and compute the errors $\Delta T_F = T^{\mathrm{ML}}[n] - T[n]$. The Gaussian and Cauchy kernels give the best performance. For the Gaussian kernel with $N = 1$ chemical accuracy is achieved (i.e. MAE less than 1 kcal/mol) at $N_T = 60$. Just as we saw in Ref. [29], the performance is systematically improvable with increasing number of training densities. The Laplacian kernel gives a mean absolute error (MAE) of 6.9 kcal/mol at $N_T = 100$ (still better than LDA), which improves as N_T increases. On the other hand, the performance of the wave kernel does not improve as N_T increases (see supplemental information). This indicates the form of the wave kernel is not flexible enough to fit the form of the KE functional.

B. Sparse grid

Note that the choice of N_G used in the reference calculations is needed to converge our reference energies and densities, but may be larger then the grid needed to "converge" our ML functional. As the ML model depends only on the inner product between densities, this will typically converge much faster than, e.g. Numerov's method. To demonstrate this, we define a "sparse" grid, $\{x_{s(j-1)+1} | j = 1, \ldots, N_G/s\}$, using every sth point in the grid (we only choose s such that N_G is divisible by s).

Fig. 8 shows that performance of the model is unaffected until N_G is reduced to about 10 grid points. The model is cross-validated each time, but the hyperparameters change only slightly. Thus, ML can accurately learn the KE functional with a far less complete basis than is required to accurately solve the Schrödinger equation. This is possible because we have restricted the learning problem to a simple type of potential with a limited range of possible densities and energies. The underlying dimensionality of the data is about 9, comparable to the number of parameters that determine the potential. The model needs only enough degrees of freedom in the representation of the density to distinguish between densities, but no more. Thus it is no coincidence that the minimum grid required is comparable to the dimensionality of the data (i.e. the dimensionality of the density manifold $\mathcal{M}_N$).

However, we also need a sufficiently fine grid to compute the integral in Eq. (7) to the desired accuracy. In the problem shown here, the dimensionality of the data is relatively small, and will increase for larger systems (e.g. real molecules with many degrees of freedom). In general, however, we need to consider both factors in choosing a suitable basis. But, we may be able to use a basis that is more sparse than that of the reference data, which would greatly reduce the computational cost of the method.

C. Challenge of finding density

Thus far, we have focused on the discussion of the performance of the MLA evaluated on exact densities (i.e. the functional-driven errors). However, in order for a functional to be useful, it must also predict the ground-state density. As discussed previously, an accurate functional derivative is necessary in order to solve Eq. (9) and yield an accurate density. The functional derivative of our MLA is given by:

$$\frac{\delta T^{\mathrm{ML}}[n]}{\delta n(x)} = \sum_{j=1}^{N_T} \alpha_j \frac{\delta k[n, n_j]}{\delta n(x)}, \tag{22}$$

where, for the Gaussian kernel,

$$\delta k[n, n_j]/\delta n(x) = (n_j(x) - n(x)) k[n, n_j]/\sigma^2. \tag{23}$$

In Fig. 10, we plot the functional derivative of our model compared with the exact derivative. The model displays

N	N_T	$\lambda \cdot 10^{14}$	σ	$\lvert\Delta T_F\rvert$ Mean	$\lvert\Delta T_F\rvert$ Max	$\lvert\Delta T\rvert$ Mean	$\lvert\Delta T\rvert$ Max	$\lvert\Delta E\rvert$ Mean	$\lvert\Delta E\rvert$ Max
1	40	50.	4.2	1.9	30.	15	120	5.1	32
	60	10.	1.8	0.62	11.	3.0	19	0.66	4.4
	80	54.	1.5	0.23	3.1	1.1	11	0.44	2.6
	100	4.5	1.6	0.13	3.5	1.4	16	0.41	2.3
	150	1.2	1.3	0.06	1.0	0.81	5.1	0.27	1.9
	200	1.3	1.0	0.03	0.87	0.67	10.	0.28	1.6
2	60	60.	3.0	0.46	4.8	1.79	9.9	0.73	3.6
	100	1.0	2.2	0.14	1.7	1.25	5.0	0.44	2.5
3	60	6.0	5.8	0.31	3.9	1.03	5.0	0.82	6.5
	100	1.9	2.5	0.13	1.7	1.11	8.3	0.59	3.8
4	60	0.6	14	0.46	5.4	2.44	9.5	0.93	6.3
	100	1.4	2.7	0.08	2.6	1.12	9.8	0.63	5.0
1-4	400	1.7	2.2	0.12	3.0	1.28	12.6	0.52	5.1

TABLE III. Hyperparameters and errors measured over the test set using the Gaussian kernel, for different N and N_T. The regularization strength λ and length scale of the Gaussian kernel σ is optimized with 10-fold cross validation. The functional-driven error $\Delta T_F = T^{\mathrm{ML}}[n] - T[n]$ is evaluated on the test set. Mean and max absolute errors are given in kcal/mol. $\Delta T = T^{\mathrm{ML}}[\tilde{n}] - T[n]$, gives the error in the KE evaluated on constrained optimal densities. Likewise $\Delta E = E^{\mathrm{ML}}[\tilde{n}] - E[n]$.

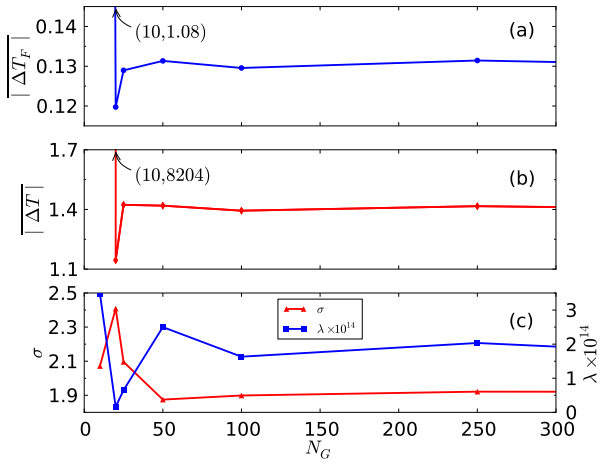

FIG. 8. The effect of using a sparse grid to represent the density on the performance of the MLA, for $N = 1$, $N_T = 100$, with the Gaussian kernel. Here (a) $\overline{|\Delta T_F|} = \overline{T^{\mathrm{ML}}[n] - T[n]}$ is the mean absolute functional-driven error of the MLA evaluated on the test set in kcal/mol, (b) $\overline{|\Delta T|} = \overline{T^{\mathrm{ML}}[\tilde{n}] - T[n]}$ gives the error of KE evaluated on constrained optimal densities in kcal/mol and (c) the corresponding re-cross validated hyperparameters λ and σ. The MAE is completely unaffected as N_G is reduced until approximately $N_G = 10$, when it jumps sharply.

a highly inaccurate functional derivative, with a huge amount of apparent "noise", as was found in Ref. [29].

What is the source of this noise? In general, if the underlying dimensionality of the data is much less than the dimensionality of J_N (which in this case is essentially infinite), ML will be unable to capture the functional derivative. The functional derivative contains information on how the KE changes along any direction, but ML cannot learn this because it only has information in directions in which it has data (i.e. along $\mathcal{M}_N$). Fig. 11 illustrates the problem: standard minimization techniques will rapidly exit the "interpolation" region in which the MLA is expected to be accurate. The MLA is only given information about how the KE changes along the density manifold $\mathcal{M}_N$. In the many dimensions orthogonal to $\mathcal{M}_N$, the MLA produces an inaccurate derivative (each of these dimensions produces a large relative error since no data exists in these directions; the sum over many dimensions creates a large total error in functional derivative). A standard gradient descent will quickly venture off of $\mathcal{M}_N$ into regions of $\mathcal{J}_N$ where the model is guaranteed to fail. Fig. 9 shows the deviation of self-consistent density if the search is not constrained to $\mathcal{M}_N$. To fix this, we further constrain the minimization in Eq. (9) to stay on $\mathcal{M}_N$. The Euler-Lagrange minimization for the ground-state density can be expressed as

$$\delta\left\{E[n] - \zeta g[n]\right\} = 0, \tag{24}$$

where g is any function that is zero on $\mathcal{M}_N$ and positive elsewhere. Thus $g[n] = 0$ implicitly defines the density manifold $\mathcal{M}$. Since any $n \in \mathcal{M}_N$ satisfies the normalization condition, the previous constraint is no longer necessary. Because the minimizing density (i.e. the ground-state density) is in $\mathcal{M}$ and thus satisfies the constraint $g[n] = 0$, Eq. 24 gives the same solution as Eq. 9. Essentially, we have vastly reduced the domain of the search from $\mathcal{J}_N$ to $\mathcal{M}_N$. To avoid confusion, we call the minimizing density of this equation the *constrained optimal*

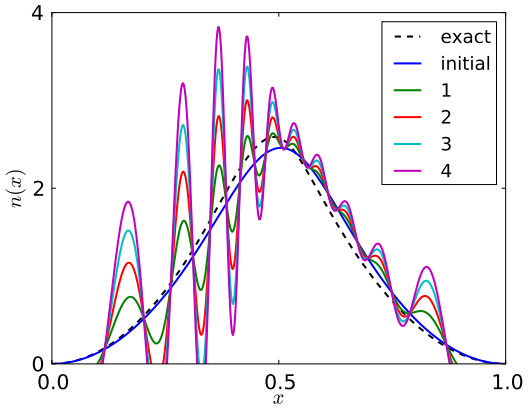

FIG. 9. The first few steps in a standard gradient descent solving the Euler equation in Eq. (9) using our MLA for the KE with $N_T = 100$ starting from a sample training density. The dashed line shows the exact self-consistent solution. The noise in the bare functional derivative quickly causes large corresponding errors in the density.

density. It may be solved self-consistently in the same sense of solving the standard Euler equation. However, the $g[n]$ which exactly gives the density manifold is unknown. In the next section, we develop an approximation which attempts to reconstruct the density manifold from the training densities.

D. Manifold reconstruction using principal component analysis

Our aim is to reconstruct $\mathcal{M}$ locally around a given density $n(x)$, which is assumed to be on the density manifold. A simple approach is to approximate $\mathcal{M}$ as locally linear, using principal component analysis (PCA) to determine the tangent space empirically from the training densities. This will work as long as there are enough training densities covering the density manifold. First, we define a weighted average density around density n:

$$\bar{n}(x) = \frac{1}{\Omega}\sum_{j=1}^{N_T}\omega_j n_j(x) \tag{25}$$

this generalized average is weighted by the function $\omega(\|n-n'\|)$ that only depends on the distance from $n(x)$ to $n'(x)$, $\omega_j = \omega(\|n-n_j\|)$, and $\Omega = \sum_{j=1}^{N_T}\omega_j$. Note that $n'(x)$ refers to the density n' evaluated at x and not the derivative of n with respect to x.

The locality of the method comes from the choice of ω. For standard PCA, the choice is $\omega(r) = \theta(R-r)$, where θ is the Heaviside function, and R is the distance from n' to the m-th nearest training density. This equally weights the nearest m training densities, and ignores all other training densities. This choice used in Ref. [29]. Here, we choose a slightly smoother weighting function:

$$\omega(r) = (1 - r/R)\theta(R - r) \tag{26}$$

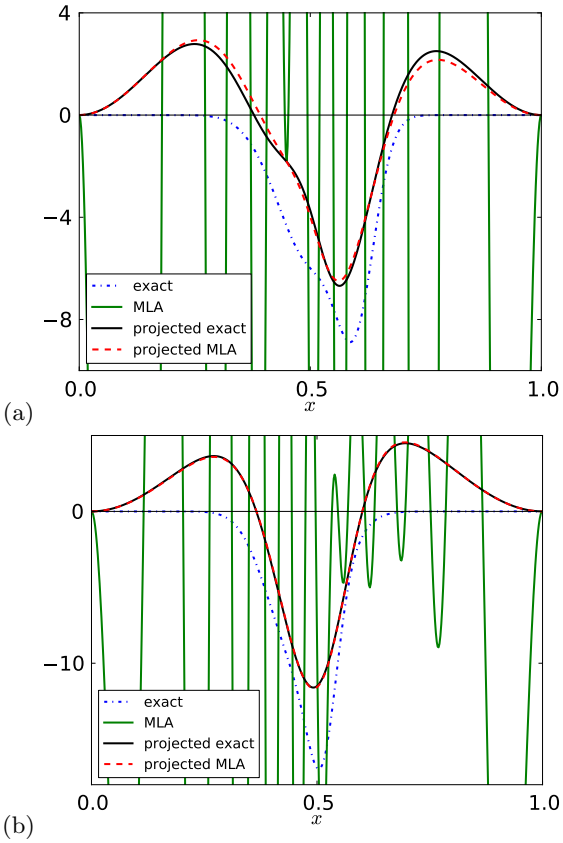

FIG. 10. The functional derivative of our MLA (green) cannot reproduce the exact derivative $v(x)$ (blue dot dashed) evaluated at the ground-state density, because this information is not contained in the data. However, both agree when projected onto the tangent of the data manifold $\mathcal{M}_N$ at n (black and red dashed). Shown for $N = 1$, for (a) $N_T = 40$ and (b) $N_T = 100$, for a typical test sample.

Next, PCA is performed by spectral analysis of the empirical covariance operator [56], based on the weighted average value around $n(x)$. We define the centered neighborhood by $\tilde{n}_j(x) = n_j(x) - \bar{n}(x)$. In this problem, densities are represented on a grid with $N_G = 500$ points, so let $\boldsymbol{n} = (n(x_1), \ldots, n(x_{N_G}))^\top$ be the vector representation of $n(x)$. The covariance matrix $\Gamma \in \mathbb{R}^{N_G \times N_G}$ is

$$\Gamma = \frac{1}{\Omega}\sum_{j=1}^{N_T}\omega_j \boldsymbol{n}_j \boldsymbol{n}_j^\top, \tag{27}$$

with eigendecomposition

$$\Gamma \boldsymbol{u}_j = \lambda_j \boldsymbol{u}_j. \tag{28}$$

The eigenvalues are ordered such that $\lambda_j > \lambda_{j+1}$. The eigenvectors $\boldsymbol{u}_j$ are called *principal components* (PCs), and give the directions of maximum variance in the data. We define the variance lost in keeping d PCs as $\eta = 1 - \sum_{j=1}^{d}\lambda_j \Big/ \sum_{j=1}^{N_G}\lambda_j$. In this case, there is little to no variance in directions orthogonal to the tangent space of

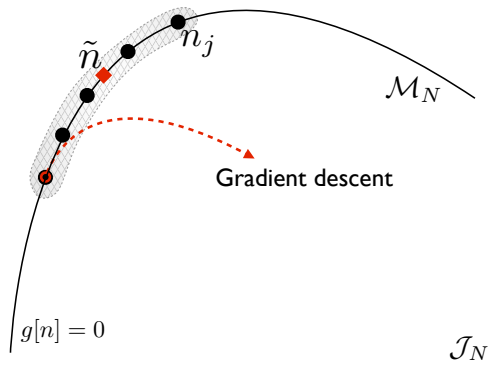

FIG. 11. Cartoon illustrating the difficulty in solving for the self-consistent density with our MLA. Pictured are the density manifold $\mathcal{M}_N$ (curved solid line), the training densities $n_j \in \mathcal{M}_N$ (black circles), and the exact self-consistent density $\tilde{n}$ (red square). Here g is a function that is identically zero on $\mathcal{M}_N$ and positive elsewhere. Thus $\mathcal{M}_N$ is defined implicitly by $g[n] = 0$. The shaded area, called the interpolation region, shows where the MLA is accurate. The solution of Eq. 9 via exact gradient descent is given by the red dashed line, which becomes unstable and soon leaves the shaded area.

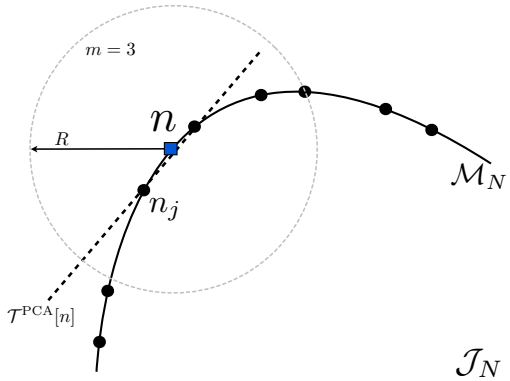

FIG. 12. Cartoon showing the density manifold $\mathcal{M}_N$ (curved line) that is contained in $\mathcal{J}_N$, the training densities n_j for $j = 1, \ldots, N_t$ (black circles). Also shown are the density $n \in \mathcal{M}$ (blue square) and the PCA approximation to tangent space of $\mathcal{M}_N$ at n, $\mathcal{T}^{\mathrm{PCA}}(n)$ (dashed line). This tangent plane is a local approximation to $\mathcal{M}_N$.

$\mathcal{M}_N$, and maximum variance in directions aligned with the tangent space. Thus, the first d PCs form a basis for the tangent space, where d is the dimensionality of the density manifold (and tangent space). The projection operator onto this basis is:

$$P[n] = \sum_{j=1}^{d} \boldsymbol{u}_j \boldsymbol{u}_j^{\top}. \tag{29}$$

The tangent space using PCA is given by

$$\mathcal{T}^{\mathrm{PCA}}[n] = \{\boldsymbol{n} \,|\, (1 - P[n])(\boldsymbol{n} - \bar{\boldsymbol{n}}) = 0\}. \tag{30}$$

Finally, we choose the PCA approximation to the constraint $g[n]$ in Eq. (24) as the squared distance from $\boldsymbol{n}$ to tangent plane $\mathcal{T}^{\mathrm{PCA}}[\boldsymbol{n}]$:

$$g^{\mathrm{PCA}}[n] = \|(1 - P[n])\tilde{\boldsymbol{n}}\|^2. \tag{31}$$

The PCA approximate density manifold $\mathcal{M}^{\mathrm{PCA}}$ is then defined implicitly by $g^{\mathrm{PCA}}[n] = 0$. The process is illustrated in Fig. 12. In the next section we develop a projected gradient descent method to solve Eq. (24).

E. Projected gradient descent algorithm

For a given ML approximation to the KE functional,

$$E^{\mathrm{ML}}[n] = T^{\mathrm{ML}}[n] + V[n], \tag{32}$$

the algorithm to minimize the functional in Eq. (24) to find a constrained optimal density is as follows (see Fig. 13). Choose an initial guess for the density, $n_0 \in \mathcal{M}_N$ (e.g., a training density):

1. Evaluate the functional derivative

$$\frac{\delta E^{\mathrm{ML}}[n]}{\delta n(x)} = \frac{\delta T_{\mathrm{S}}^{\mathrm{ML}}[n]}{\delta n(x)} + v(x). \tag{33}$$

   at $n = n_t$.

2. Compute the local PCA projection operator $P[n_t]$ from Eq. (29).

3. Project the functional derivative onto the tangent space (see Fig. 13), and take a step:

$$n_t'(x) = n_t(x) - \epsilon \hat{P}[n_t] \left. \frac{\delta E^{\mathrm{ML}}[n]}{\delta n(x)} \right|_{n=n_t}, \tag{34}$$

   where ϵ is a constant such that $0 < \epsilon \leq 1$. If convergence is unstable, reduce ϵ, trading stability for speed of convergence.

4. To ensure the constraint remains satisfied, we subtract the (weighted) mean of the training densities in the local neighborhood:

$$n_{t+1}(x) = n_t'(x) - (1 - \hat{P}[n_t'])(n_t' - \bar{n}[n_t']). \tag{35}$$

We iterate these two steps until convergence is achieved. We measure convergence by setting a maximum iteration step and tolerance threshold. If the total energy difference is smaller than tolerance within max iteration step, the density is converged. If no solution is found, ϵ is reduced.

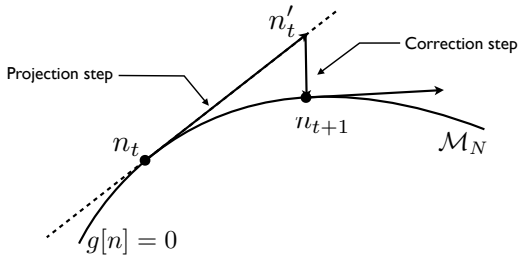

FIG. 13. Schematic of the projected gradient descent. The functional derivative is projected onto the tangent space of the data manifold $\mathcal{M}_N$ at $\tilde{n}_t$ (dashed line). Next, a step is taken along the projected functional derivative to $\tilde{n}_t'$ in the direction of lower energy. Finally, $g[n]$ is minimized orthogonal to the tangent space to ensure the minimization stays on $\mathcal{M}_N$.

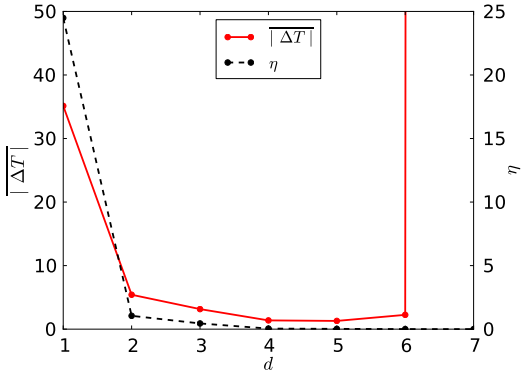

FIG. 14. The MAE, $\overline{|\Delta T|} = \overline{|T^{\mathrm{ML}}[\tilde{n}] - T[n]|}$, evaluated on 100 constrained optimal densities (in kcal/mol) compared with the variance lost η as a function of the number of PCs d in the PCA projection, with $m = 20$ nearest neighbors.

F. Errors on constrained optimal densities

With this new constrained minimization procedure via a projected gradient descent, we solve for the constrained optimal density for each test sample. We report the errors in the total energy and KE relative to the exact density in Table III. In general, we expect these errors to be worse on the MLA evaluated on exact densities—they are by roughly a factor of 10. However, errors on constrained optimal densities decrease at the same rate with more training data, so an accuracy of 1 kcal/mol in KE is achieved with 150 training samples for $N = 1$, now on constrained optimal densities. Additionally, errors are of similar magnitude for multiple particles. In the last row of Table III, we combine the training data from each N (100 training densities per N value) into one model. This combined MLA gives roughly the same error as each individual model. This is because, due to the locality of the Gaussian kernel, the training densities from each N are well separated (orthogonal in feature space) and the individual models are unaffected.

In the projected gradient descent, there are two PCA parameters that must be chosen: m, the number of nearest neighbors and d, the number of PCs to form the projection operator. Fig. 14 shows the MAE evaluated by the constrained optimal density and variance lost as a function of the number of PCs d with $m = 20$. The MAE decreases initially as d increases as more PCs capture the local structure of the density manifold. As can be seen, $d = 4$ or 5 gives an optimal reconstruction of the tangent space of the manifold. As d increases further, the noise that was removed is re-introduced into the projection, causing the gradient descent algorithm to fail. For $d = 7$, many of the constrained searches do not converge. Table IV reports the errors of the model evaluated on constrained optimal densities for $N_T = 40$ and $N_T = 100$, giving a rough optimization of the PCA parameters. Although the potential which generates $\mathcal{M}_N$ has 9 parameters in this case, we observe that the optimal choice of d is only 4. This is because the data used to build the model is only a small fraction of $\mathcal{M}_N$. If we do not sample all relevant directions on $\mathcal{M}_N$, then the model cannot learn the functional derivative in those directions. The PCA projection will compensate by removing those directions. Thus, the effectiveness of our method depends on the sampling on the manifold.

(a)

d \ m	10	20	30
2	12 (98)	15 (100)	24 (100)
3	12 (100)	16 (100)	22 (100)
4	12 (98)	15 (100)	25 (100)
5	23000 (18)	130 (27)	(0)

(b)

d \ m	10	20	30	40
3	4.1 (99)	3.2 (100)	2.7 (99)	2.8 (100)
4	1.7 (100)	1.4 (100)	1.4 (100)	1.7 (100)
5	1.6 (100)	1.3 (100)	1.5 (100)	2.0 (100)
6	1.7 (93)	2.1 (100)	1.7 (100)	2.2 (100)

TABLE IV. The error in the KE in kcal/mol evaluated on constrained optimal densities using 100 densities for testing, with $N = 1$ for (a) $N_T = 40$ and (b) $N_T = 100$. The percentage of converged optimal densities is given in parentheses. Here m is the number of nearest neighbor densities used in PCA and d is number of PCs used in the projection.

V. CONCLUSION

In this work, we have explored in much greater detail the methods presented in Ref. [29], in which ML methods were used to directly approximate the KE of a quantum system as a functional of the electron density, and used this functional in a modified orbital-free DFT to obtain

highly accurate self-consistent densities and energies.

We used a simple model as a proof of principle, to investigate how standard methods from ML can be applied to DFT. In particular, we tested a variety of standard kernels used in ML, and have found that the Gaussian kernel gives the lowest errors (the Cauchy kernel also achieves similar performance). All cross validation schemes that were tested gave similar predictions of hyperparameters that achieved low generalization error on the test set. Our results highlight the importance of an appropriate choice of kernel, as some of the kernels tested gave strikingly bad performance. With the construction of the L^2 norm that was used in the kernels, the method is basis set independent (as long as a complete basis is used). However, the ML method is capable of learning accurate KEs using a sparse grid (i.e., an incomplete basis). Using a sparse representation for the density without losing accuracy would speed up calculations further. These results warrant further exploration and will be the subject of future work.

We explained the origin of the noise in the functional derivative and developed a constrained search over the density manifold via a modified Euler equation, effectively projecting out the noise. We also introduced a local approximation to the manifold using PCA, and solved for constrained optimal densities using a projected gradient descent algorithm. This worked well for our prototype system, yielding highly accurate constrained optimal energies and densities.

ACKNOWLEDGMENTS

The authors thank for the support from NSF Grant No. CHE-1240252 (JS, KB). KRM thanks the BK21 Plus Program by NRF Korea, DFG and the Einstein Foundation. Correspondence to Li Li and K.-R. Müller.

[1] Paul Adrien Maurice Dirac, "Quantum mechanics of many-electron systems," Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character **123**, 714–733 (1929).
[2] Walter Kohn, "Nobel lecture: Electronic structure of matter-wave functions and density functionals," Reviews of Modern Physics **71**, 1253–1266 (1999).
[3] P. Hohenberg and W. Kohn, "Inhomogeneous electron gas," Phys. Rev. B **136**, 864–871 (1964).
[4] R. M. Dreizler and E. K. U. Gross, *Density Functional Theory: An Approach to the Quantum Many-Body Problem* (Springer, 1990).
[5] Llewellen H Thomas, "The calculation of atomic fields," in *Mathematical Proceedings of the Cambridge Philosophical Society*, Vol. 23 (Cambridge Univ Press, 1927) pp. 542–548.
[6] Enrico Fermi, "Eine statistische methode zur bestimmung einiger eigenschaften des atoms und ihre anwendung auf die theorie des periodischen systems der elemente," Zeitschrift für Physik **48**, 73–79 (1928).
[7] Edward Teller, "On the stability of molecules in the thomas-fermi theory," Reviews of Modern Physics (US) **34** (1962).
[8] W. Kohn and L. J. Sham, "Self-consistent equations including exchange and correlation effects," Phys. Rev. **140**, A1133–A1138 (1965).
[9] A. D. Becke, "Density-functional exchange-energy approximation with correct asymptotic behavior," Phys. Rev. A **38**, 3098–3100 (1988).
[10] Chengteh Lee, Weitao Yang, and Robert G. Parr, "Development of the colle-salvetti correlation-energy formula into a functional of the electron density," Phys. Rev. B **37**, 785–789 (1988).
[11] John P. Perdew, Kieron Burke, and Matthias Ernzerhof, "Generalized gradient approximation made simple," Phys. Rev. Lett. **77**, 3865–3868 (1996).
[12] V.V. Karasiev and S.B. Trickey, "Issues and challenges in orbital-free density functional calculations," Computer Physics Communications **183**, 2519 – 2527 (2012).
[13] Valentin V. Karasiev, Randy S. Jones, Samuel B. Trickey, and Frank E. Harris, "Recent advances in developing orbital-free kinetic energy functionals," in *New Developments in Quantum Chemistry*, edited by José Luis Paz and Antonio J. Hernández (Transworld Research Network, Kerala, India, 2009) pp. 25–54.
[14] Fabien Tran and Tomasz A. Wesolowski, "Link between the kinetic- and exchange-energy functionals in the generalized gradient approximation," International Journal of Quantum Chemistry **89**, 441–446 (2002).
[15] Linda Hung and Emily A Carter, "Accurate simulations of metals at the mesoscale: Explicit treatment of 1 million atoms with quantum mechanics," Chemical Physics Letters **475**, 163–170 (2009).
[16] Miroslav Hodak, Wenchang Lu, and J Bernholc, "Hybrid ab initio kohn-sham density functional theory/frozen-density orbital-free density functional theory simulation method suitable for biological systems," The Journal of chemical physics **128**, 014101 (2008).
[17] C.F.v. Weizsäcker, "Zur theorie der kernmassen," Zeitschrift für Physik **96**, 431–458 (1935).
[18] Y.A. Wang and E.A. Carter, "Orbital-free kinetic-energy density functional theory," in *Theoretical Methods in Condensed Phase Chemistry*, edited by S.D. Schwartz (Kluwer, NY, 2000).
[19] V.V. Karasiev, D. Chakraborty, and S.B. Trickey, "Progress on new approaches to old ideas: Orbital-free density functionals," in *Many-Electron Approaches in Physics, Chemistry, and Mathematics*, edited by L. Delle Site and V. Bach (Springer Verlag, Kluwer, NY, to appear).
[20] V. V. Karasiev, R. S. Jones, S. B. Trickey, and Frank E. Harris, "Properties of constraint-based single-point approximate kinetic energy functionals," Phys. Rev. B **80**, 245120 (2009).
[21] V. V. Karasiev, R. S. Jones, S. B. Trickey, and Frank E. Harris, "Erratum: Properties of constraint-based single-point approximate kinetic energy functionals [phys. rev. b 80, 245120 (2009)]," Phys. Rev. B **87**, 239903 (2013).

[22] Fabien Tran and Tomasz A. Wesolowski, "Link between the kinetic- and exchange-energy functionals in the generalized gradient approximation," International Journal of Quantum Chemistry **89**, 441–446 (2002).

[23] E. Chacón, J. E. Alvarellos, and P. Tarazona, "Nonlocal kinetic energy functional for nonhomogeneous electron systems," Phys. Rev. B **32**, 7868–7877 (1985).

[24] P. García-González, J. E. Alvarellos, and E. Chacón, "Nonlocal kinetic-energy-density functionals," Phys. Rev. B **53**, 9509–9512 (1996).

[25] P. García-González, J. E. Alvarellos, and E. Chacón, "Nonlocal symmetrized kinetic-energy density functional: Application to simple surfaces," Phys. Rev. B **57**, 4857–4862 (1998).

[26] Lin-Wang Wang and Michael P. Teter, "Kinetic-energy functional of the electron density," Phys. Rev. B **45**, 13196–13220 (1992).

[27] Yan Alexander Wang, Niranjan Govind, and Emily A. Carter, "Orbital-free kinetic-energy density functionals with a density-dependent kernel," Phys. Rev. B **60**, 16350–16358 (1999).

[28] Junchao Xia, Chen Huang, Ilgyou Shin, and Emily A. Carter, "Can orbital-free density functional theory simulate molecules?" The Journal of Chemical Physics **136**, 084102 (2012).

[29] John C. Snyder, Matthias Rupp, Katja Hansen, Klaus-Robert Müller, and Kieron Burke, "Finding density functionals with machine learning," Phys. Rev. Lett. **108**, 253002 (2012).

[30] John C. Snyder, Matthias Rupp, Katja Hansen, Leo Blooston, Klaus-Robert Müller, and Kieron Burke, "Orbital-free bond breaking via machine learning," J. Chem. Phys. **139**, 224104 (2013).

[31] Klaus-Robert Müller, Sebastian Mika, Gunnar Rätsch, Koji Tsuda, and Bernhard Schölkopf, "An introduction to kernel-based learning algorithms," IEEE Trans. Neural Network **12**, 181–201 (2001).

[32] Igor Kononenko, "Machine learning for medical diagnosis: history, state of the art and perspective," Artificial Intelligence in medicine **23**, 89–109 (2001).

[33] Wei Huang, Yoshiteru Nakamori, and Shou-Yang Wang, "Forecasting stock market movement direction with support vector machine," Computers & Operations Research **32**, 2513–2522 (2005).

[34] Fabrizio Sebastiani, "Machine learning in automated text categorization," ACM computing surveys (CSUR) **34**, 1–47 (2002).

[35] Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert Müller, and O. Anatole von Lilienfeld, "Fast and accurate modeling of molecular atomization energies with machine learning," Phys. Rev. Lett. **108**, 058301 (2012).

[36] Katja Hansen, Grégoire Montavon, Franziska Biegler, Siamac Fazli, Matthias Rupp, Matthias Scheffler, O. Anatole von Lilienfeld, Alexandre Tkatchenko, and Klaus-Robert Müller, "Assessment and validation of machine learning methods for predicting molecular atomization energies," Journal of Chemical Theory and Computation **9**, 3404–3419 (2013), http://pubs.acs.org/doi/pdf/10.1021/ct400195d.

[37] Gregoire Montavon, Matthias Rupp, Vivekanand Gobre, Alvaro Vazquez-Mayagoitia, Katja Hansen, Alexandre Tkatchenko, Klaus-Robert Müller, and O Anatole von Lilienfeld, "Machine learning of molecular electronic properties in chemical compound space," New Journal of Physics **15**, 095003 (2013).

[38] Zachary D. Pozun, Katja Hansen, Daniel Sheppard, Matthias Rupp, Klaus-Robert Müller, and Graeme Henkelman, "Optimizing transition states via kernel-based machine learning," The Journal of Chemical Physics **136**, 174101 (2012).

[39] Albert P. Bartók, Mike C. Payne, Risi Kondor, and Gábor Csányi, "Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons," Phys. Rev. Lett. **104**, 136403 (2010).

[40] E. Hairer, P. Nørsett, P. Syvert Paul, and G. Wanner, *Solving ordinary differential equations I: Nonstiff problems* (Springer, New York, 1993).

[41] Min-Cheol Kim, Eunji Sim, and Kieron Burke, "Understanding and reducing errors in density functional calculations," Phys. Rev. Lett. **111**, 073003 (2013).

[42] Elliott H Lieb, "Density functionals for coulomb systems," in *Inequalities* (Springer, 2002) pp. 269–303.

[43] V.N. Vapnik, *The nature of statistical learning theory* (Springer Verlag, New York, 1995).

[44] B. Scholkopf, S. Mika, C.J.C. Burges, P. Knirsch, K.-R. Muller, G. Ratsch, and A.J. Smola, "Input space versus feature space in kernel-based methods," Neural Networks, IEEE Transactions on **10**, 1000 –1017 (1999).

[45] G. Montavon, M. Braun, T. Krueger, and K.-R. Muller, "Analyzing local structure in kernel-based learning: Explanation, complexity, and reliability assessment," Signal Processing Magazine, IEEE **30**, 62–74 (2013).

[46] James Mercer, "Functions of positive and negative type, and their connection with the theory of integral equations," Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character **209**, 415–446 (1909).

[47] Nachman Aronszajn, "Theory of reproducing kernels," Transactions of the American mathematical society **68**, 337–404 (1950).

[48] B. Schölkopf, A. Smola, and K.R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," Neural computation **10**, 1299–1319 (1998).

[49] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. (Springer, New York, 2009).

[50] A.J. Smola, B. Schölkopf, and K.-R. Müller, "The connection between regularization operators and support vector kernels," Neural Networks **11**, 637–649 (1998).

[51] Alexander Zien, Gunnar Rätsch, Sebastian Mika, Bernhard Schölkopf, Thomas Lengauer, and K-R Müller, "Engineering support vector machine kernels that recognize translation initiation sites," Bioinformatics **16**, 799–807 (2000).

[52] M.L. Braun, J.M. Buhmann, and K.-R. Müller, "On relevant dimensions in kernel feature spaces," Journal of Machine Learning Research **9**, 1875–1908 (2008).

[53] Tomaso Poggio, Sayan Mukherjee, Ryan Rifkin, Alexander Rakhlin, and Alessandro Verri, *b*, Tech. Rep. AI Memo 2001-011, CBCL Memo 198 (Massachusetts Institute of Technology, 2001).

[54] S-I Amari, N Murata, K-R Müller, M Finke, and HH Yang, "Asymptotic statistical theory of overtraining and cross-validation," IEEE Transactions on Neural Networks **8**, 985–996 (1997).

[55] S. Lemm, B. Blankertz, T. Dickhaus, and K.-R. Müller, "Introduction to machine learning for brain imaging,"

Neuroimage **56**, 387–399 (2011).
[56] Christoph Bregler and Stephen M Omohundro, *Surface learning with applications to lipreading* (International Computer Science Institute, 1994).