Understanding Kernel Ridge Regression: Common behaviors from simple functions to density functionals

Kevin Vu,[1] John Snyder,[2,3] Li Li,[1] Matthias Rupp,[4] Brandon F. Chen,[5] Tarek Khelif,[5] Klaus-Robert Müller,[2,6] and Kieron Burke[1,5]

[1]*Department of Physics and Astronomy, University of California, Irvine, CA 92697*
[2]*Machine Learning Group, Technical University of Berlin, 10587 Berlin, Germany*
[3]*Max Planck Institute of Microstructure Physics, Weinberg 2, 06120 Halle (Saale), Germany*
[4]*Institute of Physical Chemistry and National Center for Computational Design and Discovery of Novel Materials (MARVEL), Department of Chemistry, University of Basel, Klingelbergstrasse 80, CH-4056 Basel, Switzerland*
[5]*Department of Chemistry, University of California, Irvine, CA 92697*
[6]*Department of Brain and Cognitive Engineering, Korea University, Anam-dong, Seongbuk-gu, Seoul 136-713, Korea*

(Dated: May 7, 2015)

Accurate approximations to density functionals have recently been obtained via machine learning (ML). By applying ML to a simple function of one variable without any random sampling, we extract the qualitative dependence of errors on hyperparameters. We find universal features of the behavior in extreme limits, including both very small and very large length scales, and the noise-free limit. We show how such features arise in ML models of density functionals.

I. INTRODUCTION

Machine learning (ML) is a powerful data-driven method for learning patterns in high-dimensional spaces via induction. It provides a whole suite of tools for analyzing data, fitting highly non-linear functions, and dimensionality reduction [1]. Given a set of training data, ML algorithms learn via induction to predict new data. ML methods have been developed within the areas of statistics and computer science, and have been applied to a huge variety of data, including neuroscience, image and text processing, and robotics. We are interested primarily in kernel ridge regression (KRR), which is one such standard method in ML. In general, the quality of the KRR learning model performance is highly dependent on the hyperparameters chosen and the size of the training data.

ML has enjoyed had widespread success in many fields, and has recently become popular as a tool in quantum chemistry and materials science [2–10], as shown by the articles in this special issue. In many of these applications, many *ab-initio* calculations are performed, and ML is applied to various properties of the results of these calculations.

Our primary interest is in creating a much more intimate relation between ML and electronic structure calculations. Many such calculations employ density functional theory, because of its favorable balance between accuracy and computational efficiency. But all such calculations rely on some approximation of an energy component as a functional of the electronic density. In particular, we wish to explore the applications of ML to the construction of density functionals [11–15], which have focused so far on approximating the kinetic energy (KE) of non-interacting electrons. An accurate, general approximation to this could make orbital-free DFT a practical reality. However, the ML methods that have been developed are quite general and have not been tailored to account for specific details of the quantum problem. For example, it was found that KRR could yield excellent results for the KE functional, while never yielding accurate functional derivatives [11]. The development of methods for bypassing this difficulty has been important for ML in general [14]. In this context, ML provides a completely different way of thinking about electronic structure. The traditional *ab-initio* approach [16] to electronic structure involves deriving carefully constructed approximations to solving the Schrödinger equation, based on physical intuition, exact conditions and asymptotic behaviors [17]. On the other hand, ML treats the physics problem as a stochastic approximation problem and learns by example, providing limited interpolation over a specific class of systems for which training data is available.

A system of N interacting electrons with some external potential is characterized by a $3N$ coordinate wavefunction, which becomes computationally demanding for large N. In the mid 1960's, Hohenberg and Kohn proved a one-to-one correspondence between the external potential of a quantum system and its one-electron ground-state density [18], showing that all properties are functionals of the ground-state density alone, which can in principle be found from a single Euler equation for the density. Although these fundamental theorems of density functional theory (DFT) proved the existence of a universal functional, essentially all modern calculations use the Kohn-Sham (KS) scheme [19], which is much more accurate, because the non-interacting KE is found exactly by using an orbital-scheme [20]. This is far faster than traditional approaches for large N, but remains a bottleneck. If a sufficiently accurate density functional for the non-interacting electrons could be found, it could increase the size of computationally tractable systems by orders of magnitude.

The Hohenberg-Kohn (HK) theorem guarantees that

all properties of the system can be determined from the electronic density alone. The basic tenet of ML is that a pattern must exist in the data in order for learning to be possible. Thus, DFT seems an ideal case to apply ML. ML learns the underlying pattern in *solutions* to the Schrödinger equation, bypassing the need to directly solve it. The HK theorem is a statement concerning the minimal information needed to do this for an arbitrary one-body potential.

Some of us recently used ML to learn the non-interacting KE of fermions in a one-dimensional box subject to smooth external potentials [11] and of a one-dimensional model of diatomics where we demonstrated the ability of ML to break multiple bonds self-consistently via an orbital-free DFT [12]. Such KE data is effectively *noise-free*, since it is generated via deterministic reference calculations, by solving the Schrödinger equation or KS equations numerically exactly. (The limited precision of the calculation might be considered "noise," as different implementations might yield answers differing on the order of machine precision, but this is negligble.) There is no noise, in the traditional sense, as is typically associated with experimental data. Note that what is considered "noise" depends on what is considered ground truth, i.e., the data to be learned. In particular, if a single reference method is used, its error with respect to a universal functional is *not* considered noise for the ML model. A perfect ML model should, at best, precisely reproduce the single-reference calculation.

As an example, in Fig. 1 we plot a measure of the error of ML for the KE of up to 4 noninteracting spinless fermions in a box under a potential with 9 parameters (given in detail in Ref. 11), fitted for different numbers of evenly spaced training densities as a function of the hyperparameter σ (called the length scale), for fixed λ (a hyperparameter called the regularization strength) and several different number of training points N_T. The scale is logarithmic,[21] so there are large variations in the fitted error. We will give a more in-depth analysis of the model performance on this data set in a later section after we have formally defined the functions and hyperparameters involved, but for now it is still useful to observe the qualitative behaviors that emerge in the figure. Note that the curves assume roughly the same shape for each N_T over the range of σ, and that they all possess distinct features in different regimes of σ.

To better understand the behavior with respect to hyperparameters seen in Fig. 1, we have chosen in this paper to apply them to the prototypical regression problem, that of fitting a simple function of one coordinate. We also remove all stochastic elements of the procedure, by considering data points on uniform grids, defining errors in the continuum limit, etc. This is shown in Fig. 2, where we plot a measure of the error of ML for a simple function $\cos x$, fitted in the region between 0 and 1, inclusive, for several N_T (represented as values on a grid) as a function of σ. Note the remarkable similarity between the features and characteristics of the curves of this figure and those of Fig. 1 (like Fig. 1 before it, we will give a more in-depth analysis of Fig. 2 later). We explore the behavior of the fitting error as a function of the number of training parameters and the hyperparameters that are used in kernel ridge regression with Gaussian kernel. We find the landscape to be surprisingly rich, and we also find elegant simplicities in various limiting cases. After this, we will be able to characterize the behavior of ML for systems like the one shown in Fig. 1.

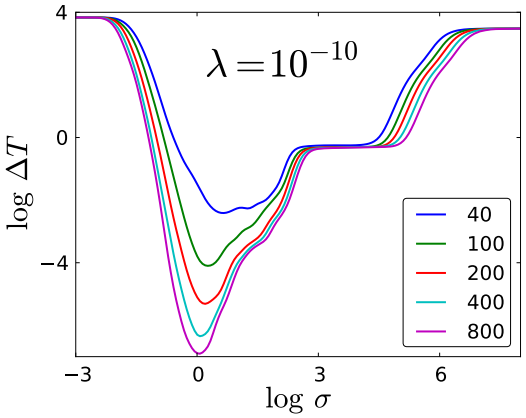


FIG. 1: The error of the model, ΔT (Hartree), for the KE of particles in a box (Section IV) as a function of σ, for fixed $\lambda = 10^{-10}$. N_T values for each curve are given in the legend.

Looking at Fig. 2, we see that the best results (lowest error) are always obtained from the middle of the curves, which can become quite flat with enough training data. Thus, any method for determining hyperparameters should usually yield a length scale somewhere in this valley. In addition, we also observe that the model error will decrease with increasing N_T. For very small length scales, all curves converge to the same poor result, regardless of the number of training points. On the other hand, notice also the plateau structure that develops for very large length scales, again with all curves converging to a certain limit. We show for which ranges of hyperparameters these plateaus emerge and how they can be estimated. We also study and explain many of the features of these curves. To show the value of this study, we then apply the same reasoning to the problem that was tackled in Refs. 11 and 13, which we showcased in Fig. 1. From the machine learning perspective our study may appear unusual as it considers properties in data and problems that are uncommon. Namely there are only a few noise free data points and all are low dimensional. Nevertheless, from the physics point of view the toy data considered reflects very well the essential properties of a highly relevant problem in quantum physics: the machine learning of DFT.

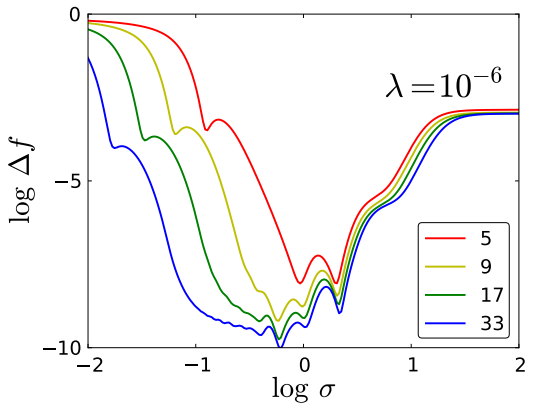


FIG. 2: Dependence of the model error (as a function of σ) when fitting $\cos x$ between 0 and 1 (Section III) for various N_T (shown in the legend) for $\lambda = 10^{-6}$.

II. BACKGROUND

In this work, we will first use ML to fit a very simple function of one variable,

$$f(x) = \cos x, \tag{1}$$

on the interval $x \in [0, 1]$. We will focus on exploring the properties of ML for this simple function before proceeding to our DFT cases. We choose a set of x-values and corresponding $f(x)$ values as the "training data" for ML to learn from. In ML, the x-values $\{x_j\}$ for $j = 1, \ldots, N_T$ are known as *features*, and corresponding f-values, $\{f_j\}$, are known as *labels*. Here N_T is the number of training samples. Usually, ML is applied with considerable random elements, such as in the choice of data points and selection of test data. In our case, we choose evenly spaced training points on the interval $[0, 1]$: $x_j = (j-1)/(N_T-1)$ for $j = 1, \ldots, N_T$. To generate the reference data, we choose 500 evenly spaced points on the interval $[0, 1]$ and evaluate $f(x)$ over these points; this is our test set.

Using this dataset, we apply KRR, which is a nonlinear form of regression with regularization to prevent overfitting [1], to fit $f(x)$. The general form of KRR is

$$f^{\mathrm{ML}}(x) = \sum_{j=1}^{N_T} \alpha_j k(x, x_j), \tag{2}$$

where α_j are the *weights* and k is the kernel (which is a measure of similarity between features). We use the Gaussian kernel

$$k(x, x') = \exp\left(-(x - x')^2/2\sigma^2\right), \tag{3}$$

a standard choice in ML that works well for a variety of problems. The hyperparameter σ is the length scale of the Gaussian, which controls the degree of correlation between training points.

The weights α_j are obtained through the minimization of the cost function

$$\mathcal{C}(\boldsymbol{\alpha}) = \sum_{j=1}^{N_T} \left(f^{\mathrm{ML}}(x_j) - f_j\right)^2 + \lambda\, \boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha}, \tag{4}$$

where

$$\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{N_T})^T \tag{5}$$

and the hyperparameter λ controls the strength of the regularization and is linked to the noise level of the learning problem. The exact solution is given by

$$\boldsymbol{\alpha} = (\boldsymbol{K} + \lambda \boldsymbol{I})^{-1} \boldsymbol{f}, \tag{6}$$

where $\boldsymbol{I}$ is the $N_T \times N_T$ identity matrix, $\boldsymbol{K}$ is the kernel matrix with elements $K_{ij} = k(x_i, x_j)$, and $\boldsymbol{f} = (f_1, \ldots, f_{N_T})^T$.

The two hyperparameters λ and σ not determined by Eq. (6) must be inferred from the data (see Section III C). σ can be viewed as the characteristic length scale of the problem being learned (the scale on which changes of f take place), as discernible from the data (and thus dependent on, e.g., the number of training samples). λ controls the leeway the model has to fit the training points. For small λ, the model has to fit the training points exactly, whereas for larger λ some deviation is allowed. Larger values of λ therefore cause the model to be smoother and vary less, i.e., less prone to overfitting. This can be directly seen in Gaussian process regression [22], a related Bayesian ML model with predictions identical to those of KRR. There, λ formally is the variance of the assumed additive Gaussian noise in values of f.

KRR is a method of interpolation. Here, we are mainly concerned with the error of the machine learning approximation (MLA) to $f(x)$ in the *interpolation region*, which in this case is the interval $x \in [0, 1]$. As a measure of this error, we define

$$\Delta f = \int_0^1 dx\, (f(x) - f^{\mathrm{ML}}(x))^2. \tag{7}$$

In the case of the Gaussian kernel, we can expand this and derive the integrals that appear analytically. To simplify the analytical process, we define

$$\Delta f_0 = \int_0^1 dx\, f^2(x), \tag{8}$$

as the benchmark error when $f^{\mathrm{ML}}(x) \equiv 0$. For the cosine function in Eq. (1),

$$\Delta f_0 = \int_0^1 dx\, \cos^2(x) = \frac{1}{2} + \frac{\sin(2)}{4} \approx 0.7273. \tag{9}$$

Now we take

$$\Delta f = \int_0^1 dx\, f(x)^2 - 2 \sum_{j=1}^{N_T} \alpha_j \int_0^1 dx\, f(x) k(x, x_j) + \sum_{i,j=1}^{N_T} \alpha_i \alpha_j \int_0^1 dx\, k(x, x_i) k(x, x_j). \tag{10}$$

where the first integral is given in Eq. (9). Next

$$\int_0^1 dx\, f(x)k(x,x_j) = \sqrt{\frac{\pi}{8}}\sigma e^{-(\gamma\sigma)^2/2}(C_j + C_j^*), \quad (11)$$

where

$$C_j = e^{i\gamma x_j}\left(\text{erf}\left(\frac{x_j - i\gamma\sigma^2}{\sigma\sqrt{2}}\right) + \text{erf}\left(\frac{1 - x_j + i\gamma\sigma^2}{\sigma\sqrt{2}}\right)\right), \quad (12)$$

erf is the error function, and C^* denotes the complex conjugate of C. The last integral is

$$\int_0^1 dx\, k(x,x_i)k(x,x_j) = \frac{\sigma\sqrt{\pi}}{2}e^{-(x_i-x_j)^2/(4\sigma^2)} \times\left(\text{erf}\left(\frac{x_i+x_j}{2\sigma}\right) - \text{erf}\left(\frac{x_i+x_j-2}{2\sigma}\right)\right). \quad (13)$$

Our goal is to characterize the dependence of the performance of the model on the size of the training data set (N_T) and the hyperparameters of the model (σ, λ). For this simple model, we discuss different regions of qualitative behavior and derive the dependence of Δf for various limiting values of these hyperparameters; we do all of this in the next few sections. In Section IV, we discuss how these results can be qualitatively generalized for the problem of using ML to learn the KE functional for non-interacting fermions in the box for a limited class of potentials.

III. ANALYSIS

We begin by analyzing the structure of Δf as a function of σ for fixed λ and N_T. Fig. 2 shows Δf as a function of σ for various N_T while fixing $\lambda = 10^{-6}$. The curves have roughly the same "valley" shape for all N_T. The bottom of the valley is an order of magnitude deeper than the walls and may have multiple local minima. These valleys are nearly identical in shape for sufficiently large N_T, which indicates that this particular feature arises in a systematic manner as N_T increases. Moreover, this central valley opens up to the left (i.e., smaller σ) as N_T increases— as the training samples become more densely packed, narrower Gaussians are better able to interpolate the function smoothly. Thus, with more training samples, a wider range of σ values will yield an accurate model.

In addition, a "cusp" will begin to appear in the region to the left of the valley, and its general shape remains the same for increasing N_T. This is another recurring feature that appears to develop systematically like the valley. For a fixed N_T, and starting from the far left, the Δf curve begins to decrease monotonically to the right, i.e., as σ increases. The cusps mark the first break in this monotonic behavior, as Δf increases briefly after reaching this local minimum before resuming its monotonic decrease for increasing σ (until this monotonicity is interrupted again in the valley region). The cusps shift to the left as N_T increases, following the trend of the valleys. This indicates that they are a fundamental feature of the Δf curves and that their appearances coincide with a particular behavior of the model as it approaches certain σ values. Note that Δf decreases nearly monotonically as N_T increases for all σ. This is as expected, since each additional training sample adds another weighted Gaussian, which should improve the fit.

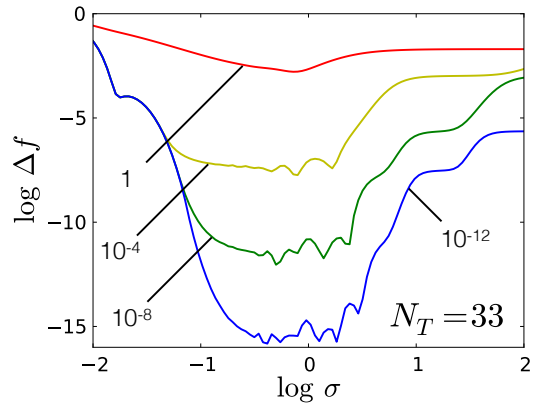
Fig. 3a again shows Δf as a function of σ, but for various λ with N_T fixed at 33. As λ decreases, Δf again decreases nearly monotonically and the central region opens up to the right (i.e., larger σ). Note that the curves for each λ coincide up to a certain σ before they split off from the rest, with the lower λ-valued curves breaking off further along to the right than those with larger λ. This shows a well-known phenomenon, namely that regularization strength λ and kernel length scale σ both give rise to regularization and smoothing [23]. Additionally, we observe the emergence of "plateau"-like structures on the right. These will be explored in detail in Section III A 2. Fig. 3b is a contour plot showing a similar situation, with the behavior of Δf for a range of σ and λ and N_T again fixed at 33.

A. Regions of qualitative behavior

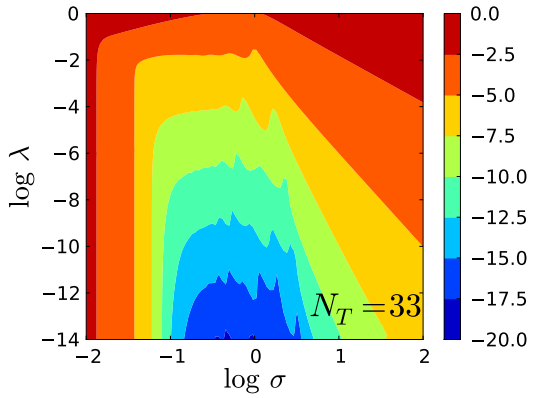
In Fig. 4, we plot Δf as a function of σ for fixed λ and N_T. The three regions labeled I, II, and III denote areas of distinct qualitative behavior. They are delineated by two arbitrary boundaries we denote by σ_s (s for small, between I and II) and σ_l (l for large, between II and III). In region I, Δf decreases significantly as σ increases. The region ends when there is significant overlap between neighboring Gaussians (i.e., when $k(x_j, x_{j+1})$ is no longer small). Region II is a "valley" where the global minimum for Δf resides. Region III begins where the valley ends and is populated by "plateaus" that correspond to $f^{\text{ML}}(x)$ assuming a polynomial form (see Section III A 2). In the following sections, we examine each region separately. In particular, we are interested in the asymptotic behavior of Δf with respect to N_T, σ and λ.

1. *Length scale too small*

The ML model for $f(x)$, given in Eq. (2), is a sum of weighted Gaussians centered at the training points, where the weights α_j are chosen to best reproduce the unknown $f(x)$. Fig. 5 shows what happens when the width of the Gaussian kernel is too small—the model is incapable of learning $f(x)$. We call this the "comb" region, as the shape of $f^{\text{ML}}(x)$ arising from the narrow Gaussians resembles a comb. In order for $f^{\text{ML}}(x)$ to accurately fit $f(x)$, the weighted Gaussians must have significant overlap. This begins when σ is on the order of the distance between adjacent training points. A corresponding general heuristic is to use a multiple (e.g., four



(a) Dependence of the model error (as a function of σ) for various λ with $N_T = 33$. The labels give the value of λ for each curve.



(b) Contour plot of the model error vs σ and λ, with $N_T = 33$. The scale of the model error is logarithmic.

FIG. 3

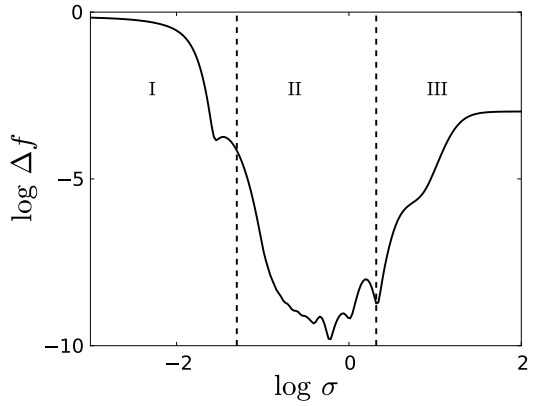


FIG. 4: Model error Δf as a function of σ for $N_T = 20$ and $\lambda = 10^{-6}$. We divide the range of σ into three qualitatively distinct regions I, II and III. The boundaries between the regions are given by the vertical dashed lines.

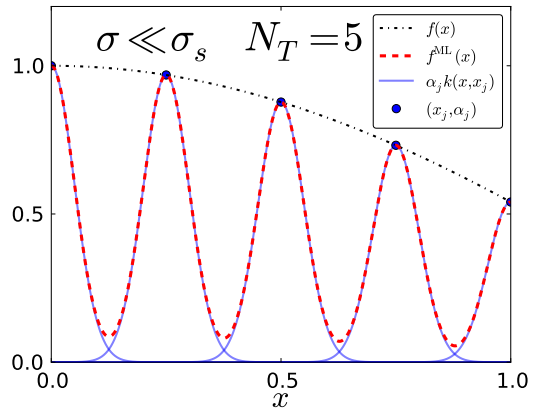


FIG. 5: Comparison of the function $f(x)$ (black dot-dashed) and the ML model $f^{\mathrm{ML}}(x)$ (red dashed), for $N_T = 5$, $\sigma = 0.05 \ll \sigma_s$ ($\sigma_s = 0.2$), and $\lambda = 10^{-6}$. When summed, the weighted Gaussians, $\alpha_j k(x, x_j)$ (blue solid), give $f^{\mathrm{ML}}(x)$. The blue dots show the location of the training points and the value of the corresponding weights. In this case, the model is in the "comb" region, when $\sigma \ll \sigma_s$. The width of the Gaussians is much smaller than the distance Δx between adjacent training points, and so the model cannot reproduce the exact function.

times) of the *median nearest neighbor distance* over the training set [12]. For equally spaced training data in one dimension, this is $\Delta x \approx 1/N_T$, so we define

$$\sigma_s = 1/N_T \tag{14}$$

to be the boundary between regions I and II. In Fig. 6, as the overlap between neighboring Gaussians becomes significant the model is able to reproduce the model well but still with significant error. Note that the common heuristics of choosing the length scale in radial basis function networks [24] are very much in line with this finding. In the comb region, Δf decreases as σ increases in a characteristic way as the Gaussians begin to fill up the space between the training points. For $\lambda \to 0$, the weights are approximately given as the values of the function at the corresponding training points:

$$\alpha_j \approx f_j, \quad \sigma \ll \sigma_s. \tag{15}$$

Thus, for small σ, the weights are independent of σ. Let $\Delta f_{\sigma \ll \sigma_s}$ be the error of the model when $\alpha_j = f_j$. This approximation, shown in Fig. 7, captures the initial decrease of Δf as σ increases, but breaks down before we reach σ_s. The qualitative nature of this decay is independent of the type of function $f(x)$, but its location and scale will depend on the specifics.

As $\sigma \to 0$ (for fixed λ and N_T), $f^{\mathrm{ML}}(x)$ becomes the sum of infinitesimally narrow Gaussians. Thus, in this limit, the error in the model becomes

$$\lim_{\sigma \to 0} \Delta f = \Delta f_0. \tag{16}$$

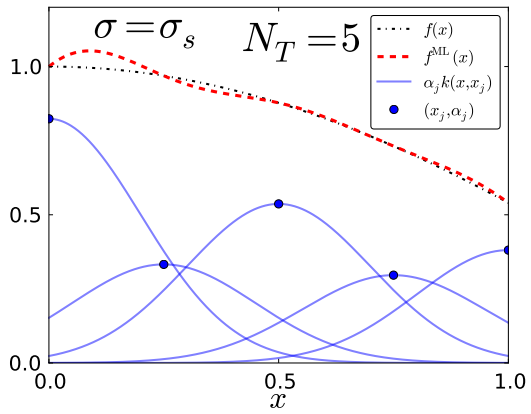


FIG. 6: Same as Fig. 5, but for $\sigma = \sigma_s = 0.2$. Here, the model is in region II, the optimum region for the model. The error in the model is very small for all x in the interpolation region. The width of the Gaussians is comparable to the size of the interpolation region.

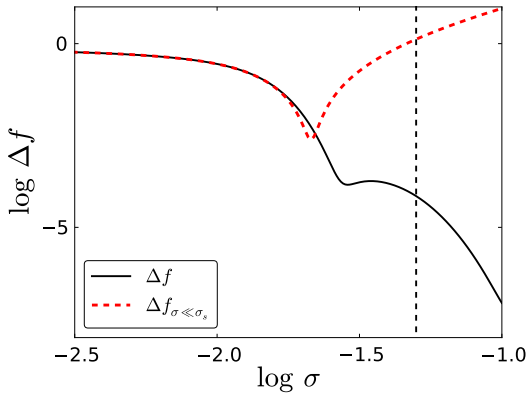


FIG. 7: Same as Fig. 4, but comparing the exact Δf against our approximation of Δf for $\sigma \ll \sigma_s$ (red dashed), where we have taken $\alpha_j \approx f_j$. The vertical dashed line shows the boundary σ_s between regions I and II. The red dashed curve breaks down before we reach σ_s, as the approximation that $\alpha_j \approx f_j$ is no longer valid.

Note that this limit is independent of λ and N_T.

Fig. 8 shows what happens when the regularization becomes too strong. (Although shown for σ in region I, the qualitative behavior is the same for any σ.) The regularization term in Eq. (4) forces the magnitude of the weights to be small, preventing $f^{\mathrm{ML}}(x)$ from fitting $f(x)$. As $\lambda \to \infty$, the weights are driven to zero, and so we obtain the same limit as in Eq. (16):

$$\lim_{\lambda\to\infty} \Delta f = \Delta f_0. \tag{17}$$

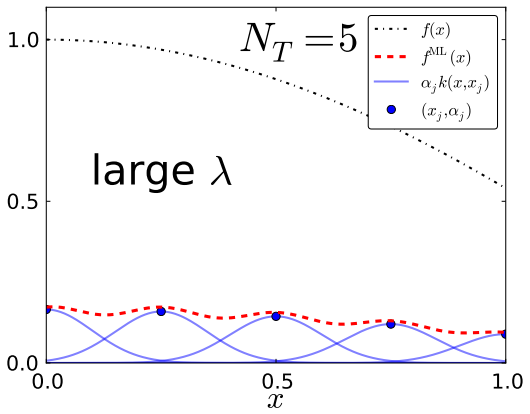


FIG. 8: Same as Fig. 5, but for $\sigma = 0.1$, and $\lambda = 5$. In this case, known as over-regularization, λ is too large, forcing the magnitude of the weights α_j to be small and preventing the model from fitting $f(x)$.

2. *Length scale too large*

We define the boundary σ_l between regions II and III as the last local minimum of Δf (with respect to σ). Thus, in region III (see Figs. 2 and 3a) Δf is monotonically increasing. As σ becomes large, the kernel functions become wide and flat over the interpolation region, and in the limit $\sigma \to \infty$, $f^{\mathrm{ML}}(x)$ is approximately a constant over $x \in [0,1]$. For small λ, the optimal constant will be the average value over the training data

$$\lim_{\lambda\to 0}\lim_{\sigma\to\infty} f^{\mathrm{ML}}(x) = \frac{1}{N_T}\sum_{j=1}^{N_T} f(x_j). \tag{18}$$

Note that the order of limits is important here: first $\sigma \to \infty$, then $\lambda \to 0$. If the order is reversed, $f^{\mathrm{ML}}(x)$ becomes the best polynomial fit of order N_T. We will show this explicitly for $N_T = 2$. For smaller σ in region III, as λ decreases, the emergence of "plateau"-like structures can be seen (see Fig. 3a). As will be shown, these flat areas correspond to the model behaving as polynomial fits of different orders. These can be derived by taking the limits $\sigma \to \infty$ and $\lambda \to 0$ while maintaining σ in certain proportions to λ, which we show in this section.

a. $N_T = 2$: In this case, the ML function is

$$f^{\mathrm{ML}}(x) = \alpha_1\, e^{-x^2/2\sigma^2} + \alpha_2\, e^{-(x-1)^2/2\sigma^2}, \tag{19}$$

and the weights are determined by solving

$$\begin{pmatrix}\alpha_1\\ \alpha_2\end{pmatrix} = \begin{pmatrix} 1+\lambda & e^{-1/2\sigma^2} \\ e^{-1/2\sigma^2} & 1+\lambda \end{pmatrix}^{-1} \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}. \tag{20}$$

The solution is

$$\alpha_1 = (f_1(1+\lambda) - e^{-1/2\sigma^2} f_2)/D, \tag{21}$$
$$\alpha_2 = (f_2(1+\lambda) - e^{-1/2\sigma^2} f_1)/D, \tag{22}$$

where $D = \det(K + \lambda I) = 1 + 2\lambda + \lambda^2 - e^{-1/\sigma^2}$. First, we expand in powers of σ as $\sigma \to \infty$, keeping up to first order:

$$\alpha_1 \approx ((f_1 - f_2) + f_1\lambda + f_2/2\sigma^2)/D, \quad (23)$$
$$\alpha_2 \approx ((f_2 - f_1) + f_2\lambda + f_1/2\sigma^2)/D, \quad (24)$$

where

$$D \approx 2\lambda + \lambda^2 + 1/\sigma^2. \quad (25)$$

Finally

$$f^{\mathrm{ML}}(x) \approx \bar{\alpha} + (\alpha_2(2x - 1) - \bar{\alpha}x^2)/2\sigma^2, \quad (26)$$

where $\bar{\alpha} = \alpha_1 + \alpha_2$. Next, we take $\lambda \to 0$. In this limit D vanishes and the weights diverge. The relative rate at which the limits are taken will affect the asymptotic behavior of the weights. The form of D suggests we take

$$\beta = \frac{1}{2\lambda\sigma^2}, \quad (27)$$

where β is a constant.

Taking $\sigma \to \infty$, we obtain a linear form:

$$f^{\mathrm{ML}}_\beta(x) = \frac{\beta f_1 + \overline{f} + \beta(f_2 - f_1)x}{\beta + 1}, \quad (28)$$

where $\overline{f} = \frac{1}{2}(f_1 + f_2)$. The parameter β smoothly connects the constant and linear plateaus. When $\beta \to 0$, we recover the constant form $f^{\mathrm{ML}}(x) = \overline{f}$; when $\beta \to \infty$, we recover the linear form $f^{\mathrm{ML}}(x) = f_1 + x(f_2 - f_1)$.

We can determine the shape of the transition between plateaus by substituting Eq. (28) for $f^{\mathrm{ML}}(x)$ into Eq. (7) for Δf. For simplicity's sake, we first define

$$h_{ij} = \int_0^1 dx\, x^i f^j(x), \quad (29)$$

since expressions of this form will show up in subsequent derivations in this work. Finally, we obtain

$$\Delta f_\beta = \frac{-2(\overline{f} + f_1\beta)h_{01}}{1 + \beta} + \frac{2\beta(f_1 - f_2)h_{11}}{1 + \beta} + \frac{(3 + 6\beta + 4\beta^2)\overline{f}^2 - f_1 f_2 \beta^2}{3(1 + \beta^2)} + h_{02}. \quad (30)$$

In Fig. 9, we compare our numerical Δf with the expansion Eq. (30) showing the transition between the linear and constant plateaus. In the case of $N_T = 2$, only these two plateaus exist. In general, there will be at most N_T plateaus, each corresponding to successively higher order polynomial fits. However, not all of these plateaus will necessarily emerge for a given N_T; as we will show, the plateaus only become apparent when λ is sufficiently small, i.e., when the asymptotic behavior is reached, and when σ and λ are proportional in a certain way similar to how we defined β. This analysis reveals the origin of the plateaus. In the series expansion for $\sigma \to \infty$, $\lambda \to 0$, certain terms corresponding to polynomial forms becomes dominant when σ and λ remain proportional.

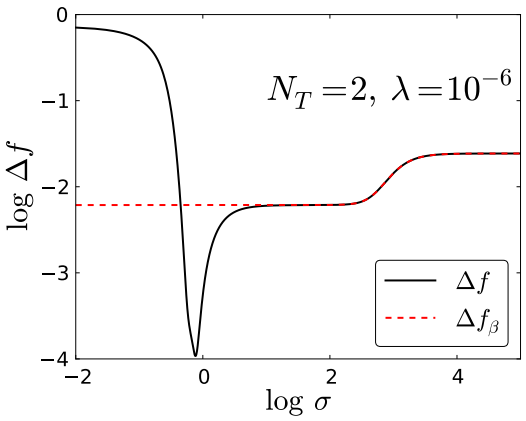


FIG. 9: Comparing the numerical Δf (black solid), for $N_T = 2$ and $\lambda = 10^{-6}$, with our asymptotic form Δf_β (red dashed) given by Eq. (30). The asymptotic form accurately recovers the behavior of Δf in the plateau regions, but fails for small σ as expected.

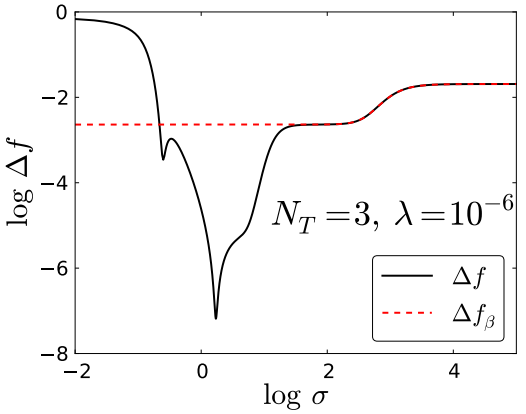


FIG. 10: Comparing the numerical Δf (black solid), for $N_T = 3$ and $\lambda = 10^{-6}$, with our asymptotic form Δf_β (red dashed). The asymptotic form accurately recovers the behavior of Δf in the plateau regions, but fails for small σ as expected.

b. $N_T = 3$: We proceed in the same manner for $N_T = 3$, using β and substituting into the analytical form of $f^{\mathrm{ML}}(x)$ for this case to obtain an expression for Δf_β (Eq. (A1), shown in the appendix). This expression is plotted in Fig. 10.

To derive the limiting value of Δf at each plateau for large N_T and small λ, we minimize the cost function Eq. (4) (which is equivalent to Eq. (7) in this limit), assuming an n-th order polynomial form for $f^{\mathrm{ML}}(x)$:

$$f^{\mathrm{ML}}(x) = \sum_{i=0}^{n} \omega_i x^i, \quad (31)$$

where the ω_i's are the polynomial coefficients that Eq. (7) must be minimized with respect to. We define c_n as the

limiting value of Δf for the n-th order plateau:

$$c_n = \lim_{N_T \to \infty} \left[\min_{\omega_i} \int_0^1 dx \, (f(x) - \sum_{i=0}^{n} \omega_i x^i)^2 \right]. \quad (32)$$

For the constant plateau, $f^{\mathrm{ML}}(x)$ assumes the constant form a; to minimize Eq. (7) with respect to a, we solve

$$\frac{d}{da} \int_0^1 dx \, (f(x) - a)^2 = 0 \quad (33)$$

for a, obtaining

$$a = \int_0^1 dx \, f(x), \quad (34)$$

so that

$$f^{\mathrm{ML}}(x) = h_{01}. \quad (35)$$

Thus, we obtain

$$c_0 = -h_{01}^2 + h_{02}. \quad (36)$$

For our case with $f(x) = \cos(x)$, $c_0 = 0.0193$.

For the linear plateau, $f^{\mathrm{ML}}(x)$ assumes the linear form $ax + b$; minimizing Eq. (7) with respect to a and b, we find that

$$f^{\mathrm{ML}}(x) = (12\, h_{11} - 6\, h_{01})x + 4\, h_{01} - 6\, h_{11}, \quad (37)$$

yielding, via Eq. (32),

$$c_1 = h_{02} - 4\left(h_{01}^2 - 3\, h_{01} h_{11} + 3\, h_{11}^2\right). \quad (38)$$

For our case with $f(x) = \cos(x)$, $c_1 = 1 \times 10^{-3}$. The same procedure yields $c_2 = 2.25 \times 10^{-6}$.

Next, we define

$$\epsilon = \frac{1}{2\lambda^{1/2}\sigma^2} \quad (39)$$

as another parameter to relate σ and λ. We choose to define this using the same motivation as for β, i.e., we examined our analytical expression for $f^{\mathrm{ML}}(x)$ and picked this parameter to substitute in order for σ and λ to remain proportional in a specific way as they approach certain limits and to see what values Δf takes for these limits (in particular, we are interested to see if we can obtain all 3 plateaus for $N_T = 3$). In doing this, we obtain an expansion analogous to that of Eq. (30) (shown as Δf_ϵ in Eq. (B1) in the appendix).

We plot this expression in Fig. 11, alongside our numerical Δf and the plateau limits, for $N_T = 3$ and varying λ. Note that the curves of the expansions are contingent on the value of λ; we do not retrieve all 3 plateaus for all of the expansions. Only the expansion curves corresponding to the smallest λ (10^{-10}, the blue curve) and second smallest λ (10^{-6}, the yellow curve) show broad, definitive ranges of σ where they take the value of each

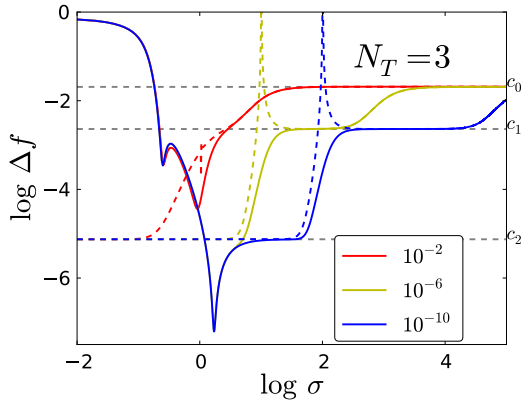


FIG. 11: The dependence of the model error on σ, for $N_T = 3$ and varying λ. The solid curves are numerical; the dashed curves are expansions derived by using our expression for ϵ in Eq. (39) and substituting into $f^{\mathrm{ML}}(x)$ in Eq. (7). The legend gives the colors (for both the dashed and solid curves) corresponding to each λ.

of the 3 plateaus (for the dashed blue curve, this is evident for c_1 and c_2; the curve approaches c_0 for larger σ ranges not shown in the figure), suggesting a specific proportion between σ and λ is needed for this to occur. For the solid numerical curves, only the blue curve manifests all 3 plateaus (like its expansion curve counterpart, it approaches c_0 for larger σ ranges not shown); the other two do not obtain all 3 plateaus, regardless of the range of σ (the solid red curve does not even go down as far as c_2). However, there appears to be a singularity for each of the expansion curves (the sharp spikes for the dashed curves) at certain values of σ (ϵ) depending on λ. This singularity emerges because our substitution of ϵ leads to an expression with ϵ in the denominator of our Δf analytical form, which naturally has a singularity for certain values of ϵ depending on λ. Following the precedent set for $N_T = 2$ and $N_T = 3$, we can proceed in the same way for larger N_T and perform the same analysis, where we expect to find higher order plateaus and the same behavior for limiting values of the parameters, including specific plateau values for Δf when σ and λ are varied with respect to each other in certain ways analogous to that of the previous cases. We would like to remark that plateau-like behaviors are well-known in statistical (online) learning in neural networks [25]. However, those plateaus are distinct from the plateau effects discussed here since they correspond to limits in the (online) learning behavior due to symmetries [26, 27] or singularities [28, 29] in the model.

3. *Length scale just right*

In the central region (see Fig. 4), Δf as a function of σ has the shape of a valley. The optimum model, i.e., the model which gives the lowest error Δf, is found in this

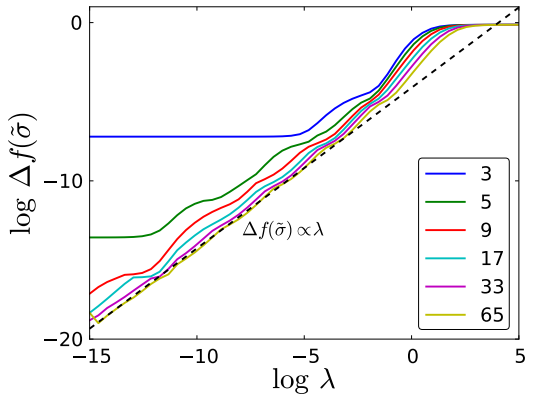


FIG. 12: The dependence of $\Delta f(\tilde{\sigma})$ on λ for various N_T. Here, $\tilde{\sigma}$ minimizes Δf for fixed N_T and λ. N_T values for each curve are given in the legend. The dashed line shows a linear proportionality between $\Delta f(\tilde{\sigma})$ and λ.

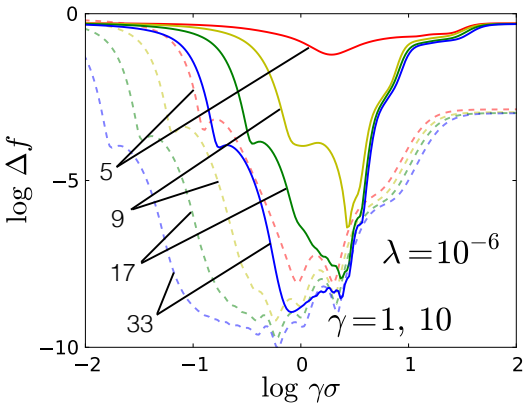


FIG. 13: Same as Fig. 2, except with σ replaced by $\gamma\sigma$, where $\gamma = 10$ for the solid curves and $\gamma = 1$ for the dashed curves. The labels give the value of N_T for each curve.

region. For fixed N_T and λ, we define the σ that gives the global minimum of Δf as $\tilde{\sigma}$. In Fig. 12, we plot the behavior of $\Delta f(\tilde{\sigma})$ as a function of λ. Again, we observe three regions of different qualitative behavior. For large λ, we over-regularize (as was shown in Fig. 8), giving the limiting value Δf_0 in Eq. (17). For moderate λ, we observe an approximately linear proportionality between $\Delta f(\tilde{\sigma})$ and λ:

$$\Delta f(\tilde{\sigma}) \propto \lambda. \tag{40}$$

However, for small enough λ, there is vanishing regularization

$$\boldsymbol{\alpha}_{\rm NF} = \lim_{\lambda\to 0}(\boldsymbol{K} + \lambda \boldsymbol{I})^{-1}\boldsymbol{f}, \tag{41}$$

yielding the *noise-free* limit of the model:

$$f^{\rm ML}_{\rm NF}(x) = \sum_{j=1}^{N_T} \alpha_{{\rm NF}\,j} k(x, x_j). \tag{42}$$

In this case (for the Gaussian kernel), this limit exists for all σ. The error of the noise-free model is

$$\Delta f_{\rm NF} = \lim_{\lambda\to 0} \Delta f. \tag{43}$$

B. Dependence on function scale

We now introduce the parameter γ into our simple one variable function, so that Eq. (1) becomes

$$f(x) = \cos(\gamma x). \tag{44}$$

For large values of γ, Eq. (44) becomes highly oscillatory; we extend our analysis here in order to observe the behavior of the model in this case.

Fig. 13 shows Δf as a function of $\gamma\sigma$ for various N_T while fixing $\lambda = 10^{-6}$ and γ=10 (solid lines), $\gamma = 1$ (dashed lines). This is the same as that of Fig. 2, except with the additional γ parameter. This figure demonstrates that the qualitative behaviors we observed in Fig. 2 persist with the inclusion of the γ parameter, complete with the characteristic "valley" shape emerging in the moderate σ region for each N_T. Similarly, we see that Δf decreases nearly monotonically for increasing N_T for all $\gamma\sigma$, while opening up to the left as the Gaussians are better able to interpolate the function. The cusps, though not as pronounced, are still present to the left of the valleys, and their general shapes remain the same for increasing N_T.

Fig. 14 shows Δf as a function of $\gamma\sigma$ for various λ while fixing $N_T = 33$ and γ=10 (solid lines), $\gamma = 1$ (dashed lines). This is the same as Fig. 3a, except with the γ parameter included. Like in Fig. 3a, as λ decreases Δf decreases nearly monotonically. The same qualitative features still hold, including the splitting-off of each lower-valued λ curve further along σ.

Next, we look at how the optimal model depends on N_T. In Fig. 15, we plot $\Delta f(\tilde{\sigma})$ as a function of N_T, for various γ. For small N_T, there is little to no improvement in the model, depending on γ. For large γ, $f^{\rm ML}(x)$ is rapidly varying and the model requires more training samples before it can begin to accurately fit the function. At this point, $\Delta f(\tilde{\sigma})$ decreases as N_T^{-c}, where $c \approx 27$ is a constant independent of γ. This fast learning rate drops off considerably when Δf is on the order of λ (i.e., at the limit of machine precision), and $\Delta f(\tilde{\sigma})$ levels off (as λ corresponds to the leeway the model has for fitting training $f(x)$ values, i.e., to the accuracy with which the model can resolve errors during fitting, it cannot improve the error much beyond this value). In fact, it is known that the learning rate in the asymptotic limit is $1/N_T$ for faithful models (i.e., models that capture the structure of the data) and $1/\sqrt{N_T}$ for unfaithful models [1, 30]. However, before the regularization kicks in Δf is approximately the noise-free limit $\Delta f_{\rm NF}$. If the noise-

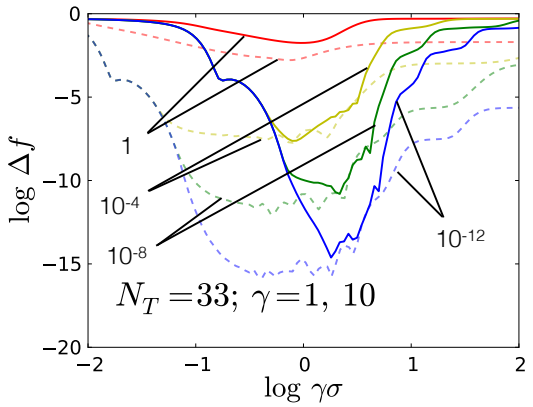


FIG. 14: Same as Fig. 3a, except with σ replaced by $\gamma\sigma$, where $\gamma = 10$ for the solid curves and $\gamma = 1$ for the dashed curves. The labels give the value of λ for each curve.

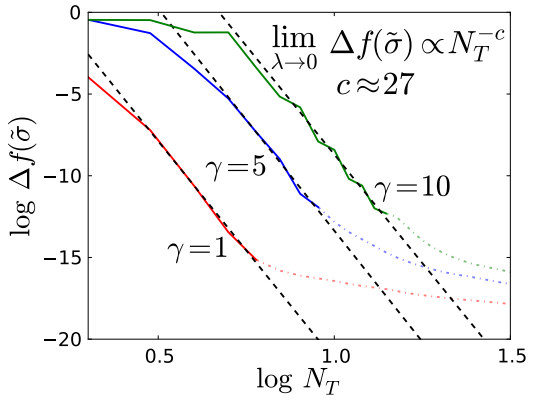


FIG. 15: The dependence of $\Delta f(\tilde{\sigma})$ on N_T for various γ. Here, $\tilde{\sigma}$ minimizes Δf for fixed N_T and λ. The solid portion of the line represents the limit at $\lambda \to 0$ (the noise-free curve), while the dot-dashed continuation shows the decay for finite λ ($\lambda = 10^{-14}$ is shown here). For large enough N_T and $\lambda \to 0$, Δf has the asymptotic form given approximately by the linear fit here (dashed line). Note that, although this asymptotic form is independent of γ, for larger γ the asymptotic region is reached at larger N_T.

free limit were taken for all N_T, it appears that $\Delta f(\tilde{\sigma})$ would decrease continually at the same learning rate:

$$\Delta f_{\rm NF} \propto N_T^{-c}. \tag{45}$$

The learning rate here resembles the error decay of an integration rule, as our simple function is smooth and can always be approximated locally by a Taylor series expansion with enough points on the interval. However, the model here uses an expansion of Gaussian functions instead of polynomials of a particular order, and the error decays much faster than a standard integration rule such as Simpson's, which decays as N_T^{-4} in the asymptotic limit. Additionally, Eq. (45) is independent of γ since, for large enough N_T, the functions appear no more complex locally. The larger y-intercepts for the larger γ curves in Fig. 15 arise due to the larger number of points needed to reach this asymptotic regime, so the errors should be comparatively larger.

C. Cross-validation

In previous works (Refs. 11 and 13) applying ML to DFT, the hyperparameters of the model were optimized in order to find the best one, i.e., we needed to find the hyperparameters such that the error for the model is minimal on the entire test set, which has not been seen by the machine in training [9]. We did this by using cross-validation, a technique whereby we minimize the error of the model with respect to the hyperparameters on a partitioned subset of the data we denote as the *validation set*. Only after we have chosen the optimal hyperparameters through cross-validation do we test the accuracy of our model by determining the error on the test set. We focus our attention on leave-one-out cross-validation, where the training set is randomly partitioned into N_T bins of size one (each bin consisting of a distinct training sample). A validation set is formed by taking the first of these bins, while a training set is formed from the rest. The model is trained on the training set, and optimal hyperparameters are determined by minimizing the error on the singleton validation set. This procedure is repeated for each bin, so N_T pairs of optimal hyperparameters are obtained in this manner; we take as our final optimal hyperparameters the median of each hyperparameter on the entire set of obtained hyperparameters. The generalization error of the model with optimal hyperparameters will finally be tested on a test set, which is inaccessible to the machine in cross-validation.

Our previous works [11–13] demonstrated the efficacy of cross-validation in producing an optimal model. Our aim here is to show how this procedure optimizes the model for our simple function on evenly-spaced training samples. We have thus far trained our model on evenly spaced points on the interval $[0, 1]$: $x_j = (j-1)/(N_T-1)$ for $j = 1, \ldots, N_T$. We want to compare how the model error determined in this way compares to the model errors using leave-one-out cross-validation to obtain optimal hyperparameters. In Fig. 16, we plot the model error over a range of σ values (we fix $\lambda = 10^{-6}$ and we use $N_T = 9$ and $N_T = 33$; compare this with Fig. 2). For each N_T, we perform leave-one-out cross-validation (using our fixed λ so that we obtain optimal σ), yielding N_T optimal σ values; we plot the model errors for each of these σ. We also include the global minimum error $\Delta f(\tilde{\sigma})$ for each N_T to show how they compare to the errors for the optimal σ. Looking at Fig. 16, we see that the optimal σ values all yield errors near $\Delta f(\tilde{\sigma})$ and within the characteristic "valley" region, demonstrating that leave-one-out cross-validation indeed optimizes our

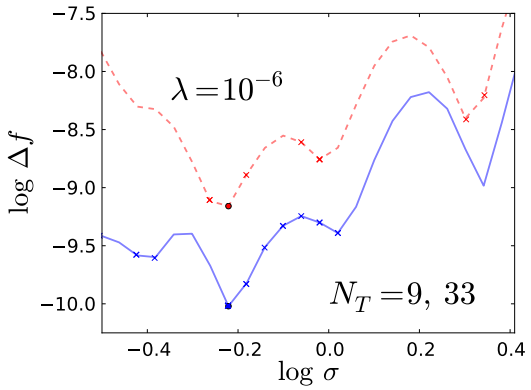


FIG. 16: The dependence of Δf on σ, for $\lambda = 10^{-6}$ and $N_T = 9$ (curve shown in red with dashed lines) and 33 (curve shown in blue with solid lines). The crosses denote Δf for the optimized σ values found from performing leave-one-out cross-validation (some of these are degenerate, so there are less than N_T distinct crosses shown), while the dots denote $\Delta f(\tilde{\sigma})$, the global minimum of Δf over σ (the crosses and dots are matched in color with the curves for each N_T).

model. With this close proximity in error values established, we can thus reasonably estimate the error of the model for the optimal σ (for a given λ) by using $\tilde{\sigma}$.

IV. APPLICATION TO DENSITY FUNCTIONALS

A canonical quantum system used frequently to explore basic quantum principles and as a proving ground for approximate quantum methods is the particle in a box. In this case, we confine one fermion to a 1d box with hard walls at $x = 0, 1$, with the addition of the external potential $v(x)$ in the interval $x \in [0, 1]$. The equation that governs the quantum mechanics is the familiar one-body Schrödinger equation in atomic units

$$\left(-\frac{1}{2}\frac{\partial^2}{\partial x^2} + v(x)\right)\phi(x) = \epsilon\phi(x). \tag{46}$$

A solution of this equation gives the orbitals $\phi_j(x)$ and energies ϵ_j. For one fermion, only the lowest energy level is occupied. The total energy is $E = \epsilon_1$, the potential energy is

$$V = \int dx\, n(x)v(x) \tag{47}$$

(where $n(x) = |\phi(x)|^2$ is the electron density), and the KE is $T = E - V$. In the case of one particle, the KE can be expressed exactly in terms of the electron density, known as the von Weizsäcker functional [31]

$$T^{\mathrm{W}} = \int dx\, \frac{n'(x)^2}{8n(x)}, \tag{48}$$

where $n'(x) = dn/dx$. Our goal here in this section is not to demonstrate the efficacy of ML approximations for the KE in DFT (which is the subject of other works [11, 12]), but rather to study the properties of the ML approximations with respect to those applications.

We choose a simple potential inside the box,

$$v(x) = -D\sin^2 \pi x, \tag{49}$$

to model a well of depth D, which has also been used in the study of semiclassical methods [32]. To generate reference data for ML to learn from, we solve Eq. (46) numerically by discretizing space onto a uniform grid, $x_j = (j-1)/(N_G - 1)$, for $j = 1, \ldots, N_G$, where N_G is the number of grid points. Numerov's method is used to solve for the lowest energy orbital and its corresponding eigenvalue. We compute T and $n(x)$, which is represented by its values on the grid. For a desired number of training samples N_T, we vary D uniformly over the range $[0, 100]$, inclusive, generating N_T pairs of electron densities and exact KEs. Additionally, a test set with 500 pairs of electron densities and exact KEs is generated.

As in the previous sections, we use KRR to learn the KE of this model system. The formulation is identical to that of Ref. 11:

$$T^{\mathrm{ML}}[n] = \sum_{j=1}^{N_T} \alpha_j k[n, n_j], \tag{50}$$

where k is the Gaussian kernel

$$k[n, n'] = \exp(-\|n - n'\|^2/(2\sigma^2)), \tag{51}$$

and

$$\|n - n'\|^2 = \Delta x \sum_{j=1}^{N_G} (n(x_j) - n'(x_j))^2, \tag{52}$$

where $\Delta x = 1/(N_G - 1)$ is the grid spacing. The weights are again given by Eq. (6), found by minimizing the cost function in Eq. (4).

In analogy to Eq. (7), we measure the error of the model as the total squared error integrated over the interpolation region

$$\Delta T = \int_0^{100} dD\, (T^{\mathrm{ML}}[n_D] - T[n_D])^2, \tag{53}$$

where n_D is the exact density for the potential with well depth D, and $T[n_D]$ is the exact corresponding KE. We approximate the integral by Simpson's rule evaluated on D sampled over the test set (i.e., 500 values spaced uniformly over the interpolation region). This sampling is sufficiently dense over the interval to give an accurate approximation to ΔT.

In Fig. 17, we plot ΔT as a function of the length scale of the Gaussian kernel, σ, for various training set sizes N_T. Clearly, the trends are very similar to Fig. 2: the transition σ_s between the regions I and II becomes

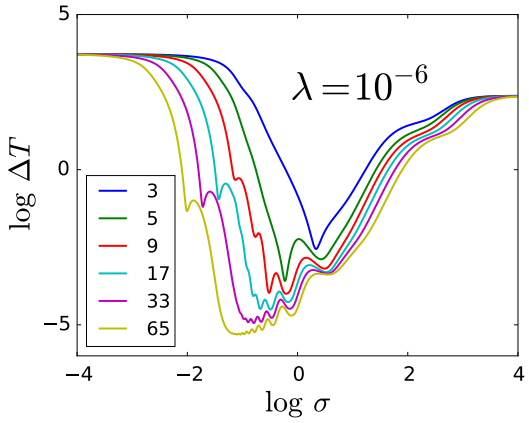


FIG. 17: The error of the model, ΔT (Hartree), as a function of σ, for fixed $\lambda = 10^{-6}$. N_T values for each curve are given in the legend.

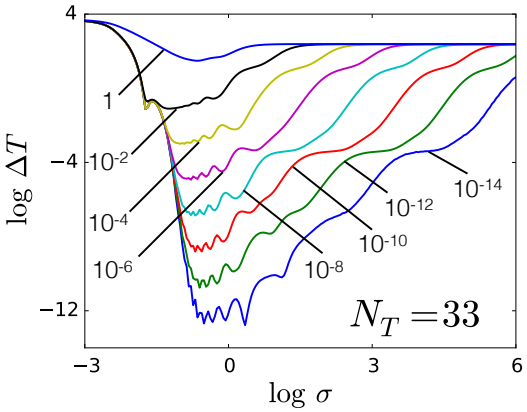


FIG. 18: The error of the model, ΔT (Hartree), as a function of σ, for various λ with $N_T = 33$. The labels give the value of λ for each curve.

smaller as N_T increases, the valley in region II widens, and region III on the right remains largely unchanged. The dependence of σ_s on N_T appears to follow the same power law $\sigma_s \propto N_T^p$, but the value of p is different in this case. A rough estimate yields $p \approx -0.8$, which is similar to $p = -1$ for the simple cosine function explored in the previous sections, but the details will depend on the specifics of the data.

Similarly, Fig. 18 shows the same plot but with N_T fixed and λ varied. Again, the same features are present as in Fig. 3a, i.e., three regions with different qualitative behaviors. In region I, ΔT has the same decay shape as the kernel functions (Gaussians) begin to overlap significantly, making it possible for the regression to function properly and fit the data. For large values of the regularization strength λ, the model over-regularizes, yielding high errors for any value of σ. As λ decreases, the weights are given more flexibility to conform to the shape of KE functional. Using the same definition for the estimation of σ_s in Eq. (14), the median nearest neighbor distance over this training set gives $\sigma_s = 0.019$. We then have $\log \sigma_s = -1.72$, which matches the boundary between regions I and II in Fig. 18. In region III, the same plateau features emerge for small λ. Again, these plateaus occur when polynomial forms of the regression model become dominant for certain combinations of the parameters σ, λ, and N_T.

From Eq. (16) and Eq. (17), we showed that the model error will tend to the benchmark error while $\sigma \to 0$ or $\lambda \to \infty$. Similarly to Eq. (8), we can also define the benchmark error when $T^{\mathrm{ML}}[n] \equiv 0$ for this data set as

$$\Delta T_0 = \int_0^{100} dD\ T^2[n_D]. \tag{54}$$

Evaluating the above integral numerically on the test set, we obtain $\log \Delta T_0 = 3.7$. This matches the error when $\sigma \to 0$ in Fig. 17 and Fig. 18.

We define the σ that gives the global minimum of ΔT as $\tilde{\sigma}$; similarly to Fig. 12, we plot the optimal model error $\Delta T(\tilde{\sigma})$ as a function of λ in Fig. 19. For large λ, we overregularize the model; the model error tends to the benchmark error in Eq. (54). For moderate λ, we observe the same linear proportionality $\Delta T(\tilde{\sigma}) \propto \lambda$ as in Fig. 12.

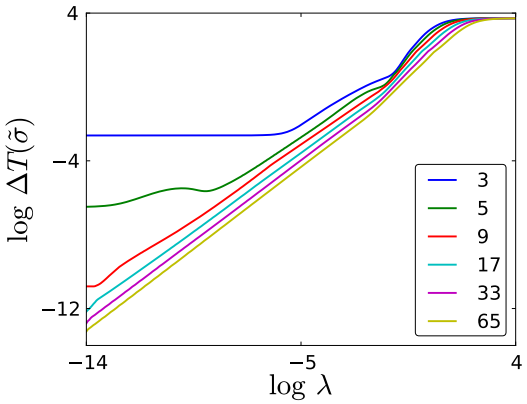


FIG. 19: The dependence of $\Delta T(\tilde{\sigma})$ on λ for various N_T. Here, $\tilde{\sigma}$ minimizes ΔT for fixed N_T and λ. N_T values for each curve are given in the legend.

In this toy system, the prediction of the KE from KRR models shares properties similar to those that we explored in learning the 1d cosine function. Now we will consider up to 4 noninteracting spinless fermions under a potential with 9 parameters as reported in Ref. 11.

$$v(x) = -\sum_{i=1}^{3} a_i \exp\left[-(x-b_i)^2/(2c_i^2)\right]. \tag{55}$$

These densities are represented on $N_G = 500$ evenly spaced grid points in $0 \le x \le 1$. Here a model is built using $N_T/4$ pairs of electron densities and exact KEs for each particle number $N = 1, 2, 3, 4$, respectively. Thus, the size of the training set is N_T. 1000 pairs of electron densities and exact KEs are generated for each N, so the

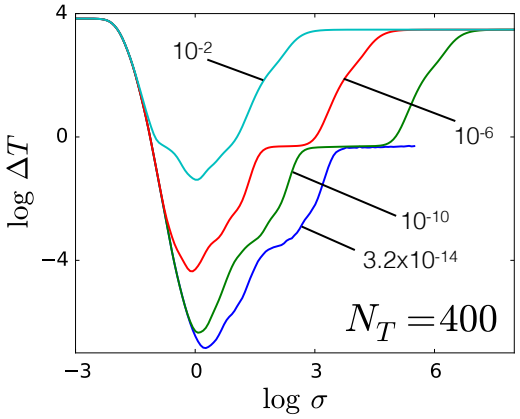


FIG. 20: Dependence of the model error (as a function of σ) for various λ. The labels give the value of λ for each curve. The $\lambda = 3.2 \times 10^{-14}$ curve is not plotted for $\log \sigma > 5.5$ due to the numerical instability that occurs when λ is small for large σ.

size of the test set is $S = 4000$. Since there are 9 parameters in the potential, we cannot define the error as an integral, so we use summation instead. Thus, the error on the test set is defined as the mean square error (MSE) on the test densities

$$\Delta T = \sum_{j=1}^{S} (T^{\mathrm{ML}}[n_j] - T[n_j])^2 / S. \tag{56}$$

Fig. 1 shows the error of the model as a function of σ with various N_T for fixed $\lambda = 10^{-10}$. Although this system is more complicated than the previous two systems discussed in this paper, the qualitative behaviors in Fig. 1 are similar to Fig. 2 and Fig. 17. Table I in Ref. 11 only shows the model error with optimized hyperparameters for $N_T = 400$. In Fig. 20, model errors varying with a wide range of σ values are shown for 4 different values of λ.[33] The qualitative properties in Fig. 20 are similar to Fig. 3a and Fig. 18. For example, the existence of three regions with distinctly different behavior for the model error can be ascertained just like before. In region I, error curves with different λ will all tend to the same benchmark error limit when $\sigma \to 0$. The median nearest neighbor distance over this training set gives $\sigma_s = 0.022$. In Fig. 20, the boundary between region I and region II is well estimated by $\log \sigma_s = -1.66$. In region III, the familiar plateau features emerge. In region II, where σ is such that the model is optimal or close to it, we find that the model with hyperparameters $\sigma = 1.86$, $\lambda = 3.2 \times 10^{-14}$ performs the best. The MSE for this model is 1.43×10^{-7} Hartree. Another common measure of error is the mean absolute error (MAE), which is also used in Ref. 11. The MAE of this model is 1.99×10^{-4} Hartree $= 0.12$ kcal/mol. This result is consistent with the model performance reported in Ref. 11.[34]

V. CONCLUSION

In this work, we have analyzed the properties of KRR models with a Gaussian kernel applied to fitting a simple 1d function. In particular, we have explored regimes of distinct qualitative behavior and derived the asymptotic behavior in certain limits. Finally, we generalized our findings to the problem of learning the KE functional of a single particle confined to a box and subject to a well potential with variable depth. Considering the vast difference in nature of the two problems compared in this work, a 1d cosine function and the KE as a functional of the electron density (a very high-dimensional object), the similarities of the measures of error Δf and ΔT between each other are remarkable. This analysis demonstrates that much of the behavior of the model can be rationalized by and distilled down to the properties of the kernel. Our goal in this work was to deepen our understanding of how the performance of KRR depends on the parameters qualitatively, in particular in the case that is relevant for MLA in DFT, namely the one of noise-free data and high-dimensional inputs, and how one may determine *a-priori* which regimes the model lies in. From the ML perspective the scenario analyzed in this work was an unusual one: small data, virtually no noise, low dimensions and high complexity. The effects found are not only interesting from the physics perspective, but are also illuminating from a learning theory point of view. However, in ML practice the extremes that contain plateaus or the "comb" region will not be observable, as the practical data with its noisy manifold structure will confine learning in the favorable region II. Future work will focus on theory and practice in order to improve learning techniques for low noise problems.

ACKNOWLEDGMENTS

The authors would like to thank NSF Grant No. CHE-1240252 (JS, LL, KB), the Alexander von Humboldt Foundation (JS), the Undergraduate Research Opportunities Program (UROP) and the Summer Undergraduate Research Program (SURP) (KV) at UC Irvine for funding. MR thanks O. Anatole von Lilienfeld for support via SNSF Grant No. PP00P2_138932. Correspondence should be sent to LL.

Appendix A: Δf_β for $N_T = 3$

$$\Delta f_\beta = h_{01}\left(\frac{1}{3}(-5f_1 - 2f_2 + f_3) + \frac{f_1 - f_3}{\beta + 1}\right) + \frac{2\beta\, h_{11}(f_1 - f_3)}{\beta + 1} + h_{02} + C, \tag{A1}$$

where

$$C = (\beta^2(7f_1^2 + 2f_1(4f_2 + f_3) + 4f_2^2 + 8f_2f_3 + 7f_3^2) + 36(2\beta + 1)\overline{f}^2)/(36(\beta + 1)^2), \quad \text{(A2)}$$

and where $\overline{f} = \frac{1}{3}(f_1 + f_2 + f_3)$.

Appendix B: Δf_ϵ for $N_T = 3$

$$\Delta f_\epsilon = C_1\, h_{01} + C_2\, h_{11} + C_3\, h_{21} + h_{02} + C_4, \quad \text{(B1)}$$

where

$$C_1 = ((\sqrt{\lambda}(48\overline{f} - \epsilon^2(17f_1 + 4f_2 + f_3)) - 2\epsilon(f_1(\epsilon^2 - 20) - 8f_2 + 4f_3) - 4\lambda\epsilon(f_2 + 4f_3)))/((\sqrt{\lambda} + \epsilon)(-8(\lambda + 3) + \epsilon^2 + 8\sqrt{\lambda}\epsilon)), \quad \text{(B2)}$$

$$C_2 = (2\epsilon(2\sqrt{\lambda}\epsilon(9f_1 + 2f_2 + f_3) + 3f_1\epsilon^2 - 24f_1 + 8\lambda(f_2 + 2f_3) - 4f_2\epsilon^2 + f_3\epsilon^2 + 24f_3))/((\sqrt{\lambda} + \epsilon)(-8(\lambda + 3) + \epsilon^2 + 8\sqrt{\lambda}\epsilon)), \quad \text{(B3)}$$

$$C_3 = \frac{4\epsilon\left(\epsilon(-f_1 + 2f_2 - f_3) - 12\overline{f}\sqrt{\lambda}\right)}{-8(\lambda + 3) + \epsilon^2 + 8\sqrt{\lambda}\epsilon}, \quad \text{(B4)}$$

$$C_4 = (2\sqrt{\lambda}\epsilon(\epsilon^4(75f_1^2 + 2f_1(58f_2 - 5f_3) + 48f_2^2 + 116f_2f_3 + 75f_3^2) - 480\epsilon^2(5f_1^2 + 5f_1f_2 + 2f_1f_3 + 2f_2^2 + 5f_2f_3 + 5f_3^2) + 34560\overline{f}^2) + 3\lambda(\epsilon^4(273f_1^2 + 232f_1f_2 + 226f_1f_3 + 48f_2^2 + 232f_2f_3 + 273f_3^2) - 160\epsilon^2(7f_1^2 + 15f_1(f_2 + 2f_3) + 4f_2^2 + 15f_2f_3 + 7f_3^2) + 11520\overline{f}^2) + 8\lambda^{3/2}\epsilon(\epsilon^2(40f_1^2 + 91f_1f_2 + 400f_1f_3 + 16f_2^2 + 91f_2f_3 + 40f_3^2) - 240\overline{f}(4f_1 + f_2 + 4f_3)) + 4\epsilon^6(2f_1^2 + 2f_1f_2 - f_1f_3 + 8f_2^2 + 2f_2f_3 + 2f_3^2) - 80\epsilon^4(5f_1^2 + 10f_1f_2 - 2f_1f_3 + 8f_2^2 + 10f_2f_3 + 5f_3^2) + 16\lambda^2\epsilon^2(48f_1^2 + 16f_1(f_2 + f_3) + 3f_2^2 + 16f_2f_3 + 48f_3^2) + 960\epsilon^2(7f_1^2 + 2f_1(4f_2 + f_3) + 4f_2^2 + 8f_2f_3 + 7f_3^2))/(60(\sqrt{\lambda} + \epsilon)^2(-8(\lambda + 3) + \epsilon^2 + 8\sqrt{\lambda}\epsilon)^2), \quad \text{(B5)}$$

and where $\overline{f} = \frac{1}{3}(f_1 + f_2 + f_3)$.

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

[2] 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).

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

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

[5] O. Ivanciuc, "Applications of Support Vector Machines in Chemistry," in *Reviews in Computational Chemistry*, Vol. 23, edited by Kenny Lipkowitz and Tom Cundari (Wiley, Hoboken, 2007) Chap. 6, pp. 291–400.

[6] 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).

[7] 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).

[8] 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).

[9] 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.

[10] K. T. Schütt, H. Glawe, F. Brockherde, A. Sanna, K. R. Müller, and E. K. U. Gross, "How to represent crystal structures for machine learning: Towards fast prediction of electronic properties," Phys. Rev. B **89**, 205118 (2014).

[11] 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).

[12] 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).

[13] Li Li, John C. Snyder, Isabelle M. Pelaschier, Jessica Huang, Uma-Naresh Niranjan, Paul Duncan, Matthias

Rupp, Klaus-Robert Müller, and Kieron Burke, "Understanding Machine-learned Density Functionals," submitted and ArXiv:1404.1333 (2014).
[14] John Snyder, Sebastian Mika, Kieron Burke, and Klaus-Robert Müller, "Kernels, Pre-Images and Optimization," in *Empirical Inference - Festschrift in Honor of Vladimir N. Vapnik*, edited by Bernhard Schoelkopf, Zhiyuan Luo, and Vladimir Vovk (Springer, Heidelberg, 2013).
[15] John C. Snyder, Matthias Rupp, Klaus-Robert Müller, and Kieron Burke, "Non-linear gradient denoising: Finding accurate extrema from inaccurate functional derivatives," International Journal of Quantum Chemistry **(submitted)** (2015).
[16] Ira N Levine, *Quantum chemistry*, Vol. 6 (Pearson Prentice Hall Upper Saddle River, NJ, 2009).
[17] R. M. Dreizler and E. K. U. Gross, *Density Functional Theory: An Approach to the Quantum Many-Body Problem* (Springer–Verlag, Berlin, 1990).
[18] P. Hohenberg and W. Kohn, "Inhomogeneous electron gas," Phys. Rev. **136**, B864–B871 (1964).
[19] W. Kohn and L. J. Sham, "Self-consistent equations including exchange and correlation effects," Phys. Rev. **140**, A1133–A1138 (1965).
[20] Aurora Pribram-Jones, David A. Gross, and Kieron Burke, "DFT: A Theory Full of Holes?" Annual Review of Physical Chemistry (2014).
[21] We will use log to denote $\log_{10}$ here and throughout this work.
[22] Carl Rasmussen and Christopher Williams, *Gaussian Processes for Machine Learning* (MIT Press, Cambridge, 2006).
[23] Alex J Smola, Bernhard Schölkopf, and Klaus-Robert Müller, "The connection between regularization operators and support vector kernels," Neural networks **11**, 637–649 (1998).
[24] John Moody and Christian J. Darken, "Fast Learning in Networks of Locally-tuned Processing Units," Neural Comput. **1**, 281–294 (1989).
[25] D. Saad, *On-line learning in neural networks* (Cambridge University Press, New York, 1998).
[26] D. Saad and S. Solla, "On-line learning in soft committee machines," Phys. Rev. E **52**, 4225–4243 (1995).
[27] M. Biehl, P. Riegler, and C. Wöhler, "Transient dynamics of on-line learning in two-layered neural networks," Journal of Physics A: Mathematical and General **29**, 4769–4780 (1996).
[28] Kenji Fukumizu and Shun Ichi Amari, "Local minima and plateaus in hierarchical structures of multilayer perceptrons." Neural Networks **13**, 317–327 (2000).
[29] Haikun Wei and Shun Ichi Amari, "Dynamics of learning near singularities in radial basis function networks." Neural Networks **21**, 989–1005 (2008).
[30] K-R Müller, M Finke, N Murata, K Schulten, and S Amari, "A numerical study on learning curves in stochastic multilayer feedforward networks," Neural Computation **8**, 1085–1106 (1996).
[31] C. F. von Weizsäcker, "Zur theorie der kernmassen," Zeitschrift für Physik A Hadrons and Nuclei **96**, 431–458 (1935), 10.1007/BF01337700.
[32] Attila Cangi, Donghyung Lee, Peter Elliott, Kieron Burke, and E. K. U. Gross, "Electronic structure via potential functional approximations," Phys. Rev. Lett. **106**, 236404 (2011).
[33] For large σ, if λ is small, there will be numerical instability in the computation of $(\boldsymbol{K} + \lambda \boldsymbol{I})^{-1}$. Thus, the $\lambda = 3.2 \times 10^{-14}$ curve is not plotted for $\log \sigma > 5.5$.
[34] In Ref. 11, the densities were treated as vectors. Here, we treated the densities as functions, so the length scale mentioned here is related to the length scale in Ref. 11 by a factor of $\sqrt{\Delta x} = 0.045$.