mBEEF-vdW: Robust fitting of error estimation density functionals

Keld T. Lundgaard,[1,2,*] Jess Wellendorff,[1,2] Johannes Voss,[1,2] Karsten W. Jacobsen,[3] and Thomas Bligaard[2,†]

[1]*Department of Chemical Engineering, Stanford University, Stanford, California 94305, USA*

[2]*SUNCAT Center for Interface Science and Catalysis, SLAC National Accelerator Laboratory,*
2575 Sand Hill Road, Menlo Park, California 94025, USA

[3]*Center for Atomic-scale Materials Design (CAMD), Department of Physics, Building 307,*
Technical University of Denmark, DK-2800 Kongens Lyngby, Denmark

(Received 8 April 2016; revised manuscript received 31 May 2016; published 30 June 2016)

We propose a general-purpose semilocal/nonlocal exchange-correlation functional approximation, named mBEEF-vdW. The exchange is a meta generalized gradient approximation, and the correlation is a semilocal and nonlocal mixture, with the Rutgers-Chalmers approximation for van der Waals (vdW) forces. The functional is fitted within the Bayesian error estimation functional (BEEF) framework [J. Wellendorff *et al.*, Phys. Rev. B **85**, 235149 (2012); J. Wellendorff *et al.*, J. Chem. Phys. **140**, 144107 (2014)]. We improve the previously used fitting procedures by introducing a robust MM-estimator based loss function, reducing the sensitivity to outliers in the datasets. To more reliably determine the optimal model complexity, we furthermore introduce a generalization of the bootstrap 0.632 estimator with hierarchical bootstrap sampling and geometric mean estimator over the training datasets. Using this estimator, we show that the robust loss function leads to a 10% improvement in the estimated prediction error over the previously used least-squares loss function. The mBEEF-vdW functional is benchmarked against popular density functional approximations over a wide range of datasets relevant for heterogeneous catalysis, including datasets that were not used for its training. Overall, we find that mBEEF-vdW has a higher general accuracy than competing popular functionals, and it is one of the best performing functionals on chemisorption systems, surface energies, lattice constants, and dispersion. We also show the potential-energy curve of graphene on the nickel(111) surface, where mBEEF-vdW matches the experimental binding length. mBEEF-vdW is currently available in GPAW and other density functional theory codes through Libxc, version 3.0.0.

DOI: 10.1103/PhysRevB.93.235162

I. INTRODUCTION

Kohn-Sham density functional theory (KS-DFT) [1,2] has become a nearly ubiquitous tool in materials science [3]. KS-DFT provides a framework for how to approximate the many-body problem by introducing the exchange-correlation (XC) functional, for which the exact form is unknown. The usefulness of KS-DFT therefore relies on finding good XC functional approximations. This can be accomplished by constraint satisfaction, using model systems, or empirically by fitting to experiments or higher-accuracy computations [4]. The XC functional is, however, only useful if we can use it to accurately predict material properties other than the systems used for the fitting. For empirical functional development, it is therefore necessary to use fitting methods that ensure such transferability [5,6].

Fitting an empirical functional requires the following: (1) choosing a proper model space, (2) gathering accurate and descriptive training data, and (3) selecting the optimal model within the model space that neither under- nor overfits the training data [7]. An often ignored problem is how outliers in the training data can influence the optimal model choice. In this work, we will introduce a fitting procedure with robust regression that is resilient to such outliers, and use it to produce a high-performing XC functional for heterogeneous catalysis studies.

When we apply KS-DFT predictions, we need to address the unavoidable inaccuracy due to an approximative XC functional. To this end, a framework for Bayesian error estimating functionals (BEEFs) was developed, where a functional ensemble would allow for error estimation [8]. Two BEEF family functionals were later produced, named BEEF–van der Waals (BEEF-vdW) [9] and meta-BEEF (mBEEF) [7]. These were both optimized as general-purpose functionals for surface science studies by fitting highly parameterized functional forms to training datasets including bulk properties, gas-phase molecular chemistry, and surface chemistry. The error estimating capabilities of these functionals have since been utilized in several surface science studies [10–13]. We here refine the previously used fitting procedure and fit a functional within an expanded exchange-correlation functional model space.

The model space complexity of the XC functionals is commonly classified through a functional ladder, with increased complexity at higher rungs [14]. At the lowest rung, only local electron density is used in the XC functional; next, one includes semilocal information, i.e., derivatives of the electron density; and finally, fully nonlocal information is included, first for the electron density and then for the wave functions. Higher complexity leads to higher computational cost, but allows for a more accurate functional, which can overcome the otherwise inevitable compromises between describing different material properties [7,15].

In this work, we focus on fitting on the semilocal meta-GGA rung for exchange, which uses as ingredients the density, the density gradient, and the Kohn-Sham kinetic energy

[*] keld@stanford.edu; keld.lundgaard@gmail.com
[†] bligaard@slac.stanford.edu

density. For correlation, we will also consider the nonlocal density-density overlap. Higher-rung functionals that use the nonlocal exact exchange are so computationally demanding that they become unfeasible for most heterogeneous catalysis studies [16]. For recent advancements in the development of meta-GGA functionals, both empirical and nonempirical, see Refs. [6,17–26].

The BEEF-vdW [9] functional was fitted within semilocal generalized gradient approximation (GGA) for exchange, which depends on the electronic density and its derivative. Its correlation was a fitted mixture of a local density approximation (LDA), semilocal GGA, and nonlocal correlation of Rutgers-Chalmers approximation for van der Waals forces [27,28]. The mBEEF [7] functional was fitted within the meta generalized gradient approximation (MGGA) for its exchange, which includes the kinetic energy density such that the model can distinguish between different types of electron orbital overlap [29], and it uses a GGA-type correlation.

A limited model space for the XC approximation means that the XC functional creator will make compromises between the accuracy of predicting different material properties [7]. For the BEEF functionals, the compromises can be summarized as follows. BEEF-vdW [9] has a high accuracy on chemisorption systems compared to semilocal functionals and reasonable accuracy on dispersion systems relative to other GGA-vdW functionals. However, BEEF-vdW has a lower accuracy on lattice constants and surface energies compared to the best semilocal functionals, which is generally true for most GGA-vdW functionals [9,30]. The mBEEF [7] functional has a high accuracy on both chemisorption energies and lattice constants relative to other semilocal functionals, and thereby overcomes the limits of GGA-type exchange functionals. However, its accuracy is limited on binding energies for systems where long-range dispersion is important compared to GGA-vdW functionals, which could be attributed to a lack of a van der Waals correlation term. A BEEF functional that combines the model spaces of BEEF-vdW and mBEEF is therefore a logical step forward as such a functional should be able to achieve a high accuracy on chemisorption, dispersion, and lattice constants simultaneously.

In this work, we parametrize the XC functional model space of MGGA exchange and correlation with nonlocal van der Waals correction. Within this model space, we fit a functional using a robust fitting procedure with a cost function using a product of robust loss functions for the training datasets, and regularization with a nonsmoothness penalty on the fitting coefficients. For choosing the regularization strength, we use a generalization of the bootstrap 0.632 estimating prediction error with geometric mean over datasets and hierarchical sampling. We name this functional mBEEF-vdW and we propose that it is a computationally efficient and generally applicable exchange-correlation functional for heterogeneous catalysis.

The structure of the paper is the following. In Sec. II, we present the methods used for fitting the mBEEF-vdW functional including the parameter space, the training datasets, and the model selection procedure. In Sec. III, we present the optimization of the most important variables in the fitting scheme. In Sec. IV, we present the mBEEF-vdW functional form. In Sec. V, we benchmark mBEEF-vdW against popular semilocal and nonlocal density functionals. Lastly, in Sec. VI, we summarize, discuss, and conclude.

II. METHODS

A. Parameter space

For the parametrization of the exchange-correlation energy functional, we use the flexible exchange energy parametrization introduced for mBEEF [7], while the correlation is parameterized as a mixture of correlation functionals from the literature, similar to what was done for fitting BEEF-vdW [9].

Following the usual conventions [31], we write the exchange energy from the semilocal meta generalized gradient approximation (MGGA) as an integral over the uniform electron gas exchange energy density ϵ_x^{UEG} scaled with an exchange enhancement factor F_x, hence

$$E_x = \int n\epsilon_x^{UEG}(n)F_x(n,\nabla n,\tau)d\mathbf{r}, \tag{1}$$

where $n = n(\mathbf{r})$ is the local electron density, ∇n is the density gradient, and $\tau = \frac{1}{2}\sum_{i,\sigma}|\nabla\Psi_{i,\sigma}|^2$ is the semilocal kinetic energy density, which is summed over spins σ and state labels i for the KS eigenstates $\Psi_{i,\sigma}$. Atomic units are used throughout. The enhancement factor's parameters are made dimensionless by introducing the reduced density gradient $s = |\nabla n|/(2k_F n)$ with $k_F = (3\pi^2 n)^{\frac{1}{3}}$, and the reduced kinetic energy density $\alpha = (\tau - \tau^W)/\tau^{UEG}$, where $\tau^W = |\nabla n|^2/8n$ and $\tau^{UEG} = (3/10)(3\pi^2)^{\frac{2}{3}}n^{\frac{5}{3}}$. With these definitions, the MGGA exchange enhancement factor can be expressed as a function of s and α.

For our parametrization of the MGGA exchange enhancement factor, we introduce the transformations t_s and t_α for s and α, and expand $F_x(t_s,t_\alpha)$ in products of one-dimensional Legendre polynomials $\boldsymbol{B}$ of either t_s or t_α, similar to what was done in Ref. [7]:

$$t_s(s) = \frac{2s^2}{q+s^2} - 1, \tag{2}$$

$$t_\alpha(\alpha) = -\frac{(1-\alpha^2)^3}{1+\alpha^3+\alpha^6}, \tag{3}$$

$$P_{mn} = B_m(t_s)B_n(t_\alpha), \tag{4}$$

$$F_x(s,\alpha) = \sum_{m_s=0}^{M_s-1}\sum_{m_\alpha=0}^{M_\alpha-1} a_{M_\alpha\cdot m_s+m_\alpha+1}P_{m_s,m_\alpha}, \tag{5}$$

where a_k is the kth fitting coefficient that we wish to find with $M_x = M_s \cdot M_\alpha$, hence $k \in \{1,2,\ldots,M_x\}$. With the above transformations, t_s and t_α are confined to the interval $[-1, +1]$ on which the Legendre polynomials form an orthogonal basis. For t_s, we chose $q = \kappa/\mu = 0.804/(10/81) = 6.5124$, such that the s dependency in principle could fulfill the slowly varying electron gas limit close to $s = 0$ [32]. We similarly choose the transformation t_α such that the second-order gradient expansion can be fulfilled according to Ref. [33].

For the correlation energy, we use a parametrization given by

$$E_c[n,\nabla n] = a_{LDA}E_c^{LDA} + a_{sl}E_c^{sl} + a_{nl}E_c^{nl}, \tag{6}$$

where E_c^{LDA} is the local Perdew-Wang correlation [34], E_c^{sl} is the semilocal (sl) correlation energy functional of either Perdew-Burke-Ernzerhof (PBE) [35], PBEsol [36], vPBE [33], or revised Tao-Perdew-Staroverov-Scuseria (revTPSS) [37], and E_c^{nl} is the nonlocal (nl) correlation energy from either vdW-DF [28] or vdW-DF2 [27]. The correlation coefficients take the indexes $M_x + \{1,2,3\}$ in the coefficient vector $\mathbf{a}$. The parameter space for the correlation has thereby been expanded compared to BEEF-vdW by the inclusion of a coefficient on the term for nonlocal correlation and by using independent fitting parameters on local and semilocal correlation [38].

B. Training datasets

We train the model on a subset of the datasets previously introduced and used in earlier BEEF functional studies [7,9]:

RE42 42 reaction energies that represent gas-phase chemistry [39].

CE27 27 chemisorption energies [7].

Sol54Ec Cohesive energies of 54 solids [7].

Sol58LC-dEc The derivatives of 58 cohesive energies with respect to the crystal volumes around the experimental equilibrium lattice constants, taken from the Sol58LC dataset [7].

S22×5 Noncovalent interaction of the 22 intermolecular interaction energies, with the interaction energies of the relative distances of 0.9, 1.0, 1.2, 1.5, and 2.0 [40]. The reference values have been corrected as in Ref. [9].

All DFT calculations were done in GPAW [41,42] and the computational details are the same as those of Refs. [7,9].

C. Model selection

We seek the coefficient vector that gives the best performing functional, i.e., neither under- nor overfit the training datasets. The starting point for our fitting procedure is the least sum of squares (LS), which we will extend to resolve its shortcomings. With LS, we minimize the cost function $C = (\mathbf{X}\mathbf{a} - \mathbf{y})^2$, where $\mathbf{y}$ is the target vector of length N, $\mathbf{a}$ is the coefficient vector of length P, and $\mathbf{X}$ is the design matrix of size $N \times P$. The LS cost function should, in principle, be minimized by $\mathbf{a} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$. However, $\mathbf{X}^T\mathbf{X}$ can have eigenvalues close to zero and this can lead to an overfit of the training data. The instability can be handled by adding a second term to the cost function, a so-called regularization term, which penalizes the Euclidean norm of the coefficient vector. We can write the cost function with a regularization as $C = (\mathbf{X}\mathbf{a} - \mathbf{y})^2 + \omega^2\mathbf{a}^2$, where $\omega \geqslant 0$ is a constant that scales the regularization penalty [43]. This method is referred to as ridge regression and we will refer to it as RR-LS for ridge regression with a least-sum-of-squares loss function. We can write this cost function in the form

$$C = L(\mathbf{a},D) + R(\mathbf{a},\omega), \tag{7}$$

where the loss term L provides a measure for how well the model $\mathbf{a}$ performs on the training data D, and the regularization term R is a measure for the model complexity [44,45]. The loss and regularization terms are balanced through the regularization strength ω as we saw with RR-LS. To choose the optimal regularization strength, we can use cross-validation techniques to provide a measure for the transferable accuracy of the model to systems that the model was not trained on, and optimize for this measure [45].

The RR-LS method with its regularization is superior to the LS method; however, there are a number of problems that RR-LS cannot handle well: different scales of the training data, weighing the penalty between different basis functions, and outliers in the training data. To handle these problems, we refine the RR-LS fitting procedure in the context of XC fitting similar to the previous BEEF functional fitting studies, but with some further advancements [7,9]. First, we will introduce a cost function, where we use the geometric mean to make compromises between how well we fit each training dataset. Second, we introduce a regularization term, which uses smoothness for penalizing the coefficients of the exchange enhancement factor. Third, we present a new estimate for the transferable accuracy of the fit, which we use to find the regularization strength with a generalization of the 0.632 bootstrap estimator. Fourth, we employ a robust loss function instead of LS to make the fit robust towards outliers in the dataset. And fifth, we adjust the Bayesian error estimating ensemble creation to account for the use of a robust loss function. We introduce the third and fifth points to the present study, whereas the first and second points are from Ref. [7].

1. Geometric mean loss function

The number of elements and the physical units in the datasets vary. If we treat all data points equally and use the sum of squared fitting errors over the training datasets, we will skew the solution compromise towards the datasets with more systems and larger absolute fitting errors unless we add a normalization. We can, however, avoid the need for a normalization if we use a geometric mean over the training datasets instead of the arithmetic mean. Following the procedure from mBEEF [7], we create an objective function as a product over the loss functions for the datasets multiplied by a regularization term given as

$$\Phi(a,\omega,\mathcal{W}) = \prod_i L_i(\mathbf{a})^{\mathcal{W}_i} \cdot e^{R(\mathbf{a},\omega)}, \tag{8}$$

where the ith dataset has a loss function L_i and a weight $\mathcal{W}_i$ [46]. Compared to the form in mBEEF, we include a weighing of the datasets similar to the fitting procedure for BEEF-vdW [9], which allows us to tune the compromise between the different datasets. The loss function term L could be LS or a robust loss function, as we will show later.

However, we would like to bring the optimization problem back to a linear form as this would allow for a simple optimization strategy and would make the error estimation method more straightforward. To this end, we will follow the derivation in Ref. [7]. We can take the logarithm of Φ without changing its minimum and, therefore, minimize $K = \ln(\Phi) = \sum_i \mathcal{W}_i \ln\{L_i(\mathbf{a})\} + R(\mathbf{a},\omega)$ instead of Φ. Next, we use the zero-gradient condition to find the minimum of K

by solving

$$\frac{\partial K}{\partial \mathbf{a}} = 0 = \sum_i \mathcal{W}_i \frac{\partial \ln L_i}{\partial \mathbf{a}} + \frac{\partial R}{\partial \mathbf{a}} = \sum_i \mathcal{W}_i \frac{\partial L_i}{L_i \partial \mathbf{a}} + \frac{\partial R}{\partial \mathbf{a}}, \tag{9}$$

where we have left out the dependencies of L and R on $\mathbf{a}$ and ω. The zero-gradient condition of the RR-LS cost function form in Eq. (7) is given by $\frac{\partial C}{\partial \mathbf{a}} = \frac{\partial L}{\partial \mathbf{a}} + \frac{\partial R}{\partial \mathbf{a}}$. Ignoring the dataset weights, we see that the only difference between the solution to RR-LS and Eq. (9) is the normalization by L. We can therefore linearize our product cost function of Eq. (8) as

$$\tilde{K}(\boldsymbol{a};\omega) = \sum_i \mathcal{W}_i \frac{L_i(\boldsymbol{a})}{L_i(\boldsymbol{a}_0)} + R(\boldsymbol{a};\omega) \tag{10}$$

$$= \tilde{L}(\mathbf{a},\mathbf{a}_0) + R(\boldsymbol{a};\omega), \tag{11}$$

where the optimal solution $\mathbf{a}_0$ is estimated iteratively by minimizing $\tilde{K}$ given a starting guess of $\mathbf{a}_0$. We are aware that minimizing $\tilde{K}$ can result in a suboptimal solution if the cost function of Eq. (8) has many local minima around $\mathbf{a}_0$; however, this does not seem to be a practical problem for fitting our functional as tests with different starting points resulted in the same solution.

2. *Regularization*

We use a quadratic regularization term with a Tikhonov transformation given as

$$R(\mathbf{a},\omega) = [\boldsymbol{\Gamma}(\mathbf{a} - \mathbf{a}_p)]^2, \tag{12}$$

where a_p is the origo for the regularization and $\boldsymbol{\Gamma}^2$ is the Tikhonov matrix [44]. The Tikhonov matrix is uncoupled between the basis functions for exchange and correlation, and it takes the form

$$\boldsymbol{\Gamma}^2 = \boldsymbol{\Gamma}_x^2 + \lambda_{c/x}\mathbf{I}_c + \lambda_I \mathbf{I}, \tag{13}$$

where $\boldsymbol{\Gamma}_x^2$ is the Tikhonov matrix of the exchange basis functions, $\mathbf{I}_c$ is an identity matrix over the correlation basis functions with a scaling constant $\lambda_{c/x}$, and the identity matrix $\mathbf{I}$ covers both correlation and exchange basis with a scaling constant λ_I.

For the exchange part of the Tikhonov matrix $\boldsymbol{\Gamma}_x^2$, we find the entries by calculating a two-dimensional smoothness measure, which is given as an integral over the Laplacian $\tilde{\nabla}^2$ of the basis functions $P(t_s,t_\alpha)$, hence

$$\tilde{\nabla}^2 = \frac{\partial^2}{\partial t_s^2} + \lambda_{\alpha/s}\frac{\partial^2}{\partial t_\alpha^2}, \tag{14}$$

$$\boldsymbol{\Gamma}^2_{x,mnkl} = \int_{-1}^{1}\int_{-1}^{1} dt_s\, dt_\alpha \tilde{\nabla}^2 P_{mn} \tilde{\nabla}^2 P_{kl}, \tag{15}$$

where $\lambda_{\alpha/s}$ scales the regularization penalty between polynomials in t_s and t_α. Note that $\boldsymbol{\Gamma}_x^2$ is zero for the zeroth- and first-order terms, and the additional term $\lambda_I \mathbf{I}$ is therefore included in Eq. (13) to prevent numerical instabilities.

3. *Hierarchical 0.632 bootstrap prediction error estimator*

To determine the optimal regularization strength ω and compare different loss functions, we introduce a generalization of Efron's 0.632 bootstrap estimated prediction error (EPE) [47]. We generalize the bootstrap sampling by sampling the training datasets hierarchically and by using the geometric mean over the training datasets in the bootstrap estimators.

a. The sampling procedure. We create a hierarchical bootstrap sample b in two steps: first, we sample the training datasets internally by randomly drawing with replacement, and second, we randomly draw a collection of datasets with replacement from the resampled training datasets. A bootstrap sample b will therefore only have a subset of the original training datasets, and each of these datasets will only have a portion of their data points present. We have the same number of training datasets in each sample, but these datasets vary in size and the total number of data points in each sample will therefore also vary.

b. The estimated prediction error. Following the original bootstrap 0.632 procedure, we write the estimated prediction error (EPE) as

$$\text{EPE} = \sqrt{0.368\,\text{err} + 0.632\,\text{ERR}}, \tag{16}$$

where err is the training error, which is the variance of the prediction error for all training data, and ERR is the bootstrap error, which measures the transferability through calculating the variance of bootstrap sample predictions [45].

We define the training error (err) as the weighted geometric mean of the mean squared error for each dataset, hence

$$\text{err} = \left(\prod_i \text{err}_i^{\mathcal{W}_i}\right)^{1/\sum_i \mathcal{W}_i}, \quad \text{err}_i = \frac{1}{N_i}(\mathbf{X}_i\mathbf{a} - \mathbf{y}_i)^2, \tag{17}$$

where $\mathbf{a}$ is the optimal model for all data, and i in $\mathbf{X}_i$ and $\mathbf{y}_i$ means that we take the design matrix and target vector associated with the ith dataset.

We similarly generalize the leave-one-out bootstrap estimator ERR with the geometric mean over the training datasets. For each data point j of dataset i, we calculate the mean squared error of the predictions from fitting to the bootstrap samples $\mathbf{b}$ where the data point was not present; next, we calculate the mean of these squared errors for each dataset; and finally, we calculate the weighted geometric mean over the training datasets, hence

$$\text{ERR} = \left(\prod_i \text{ERR}_i^{\mathcal{W}_i}\right)^{1/\sum_i \mathcal{W}_i}, \tag{18}$$

$$\text{ERR}_\text{i} = \frac{1}{N_i}\sum_j \frac{1}{|C^{-(i,j)}|}\sum_{b\in C^{-(i,j)}} (\mathbf{x}_{i,j}\mathbf{a}_b - y_{i,j})^2, \tag{19}$$

where $C^{-(i,j)}$ is the subset of bootstrap samples $\mathbf{b}$ of size $|C^{-(i,j)}|$ without data point (i,j), $\mathbf{x}_{i,j}$ is the jth row of the design matrix for dataset i, $y_{i,j}$ is the jth target value of dataset i, and $\mathbf{a}_b$ is the optimal solution for bootstrap sample b.

For a single dataset, the sampling method, ERR, err, and EPE all reduce to the original bootstrap 0.632 formulation.

4. *Robust loss function*

Let us revisit the least-sum-of-squares (LS) optimization criteria in the loss function. LS is the most popular loss function mainly because of its computational simplicity, rather than its optimal efficiency for regression problems with a normal distributed noise [48]. However, LS is also very sensitive to outliers in the data. If we take a single training data point and change it to an extreme value, the optimal model of LS can become useless for reproducing the rest of the training data. We might not even detect such outliers in the training data because we evaluate whether data points are outliers using a model influenced by the outliers, which creates a masking effect [49].

We could instead use a robust estimator such as the least median of squared residuals (LMS), which minimizes the median of $\{r_j^2, j = 1,\dots,N\}$, where $r_j = \mathbf{x}_j\mathbf{a} - y_j$ is residual for the jth data point [48]. Similar to LS, we find the scale estimate of LMS by the square of this criteria, which is called the median absolute deviation about zero. For the LMS, we can arbitrarily change almost 50% of the training data and the estimator will still provide a good scale estimate to the rest of the data. We therefore say that the LMS has a breakdown point of 50%, which is as good as you can do [50]. However, the LMS loss function lacks a high normal efficient, meaning that the estimate performs much worse than LS if we were to fit data fully explained by our model space, but with a normal distributed noise on the training data.

Other estimators have, however, been proven to achieve both a high breakdown point and a high normal efficiency [49]. One example is the MM-estimator, which we will use for our loss function [51]. In the following, we will present the MM-estimator and the S-estimator that is used as the first step of the MM-estimator procedure, starting with the S-estimator [52].

a. S-estimators. Both the LS and LMS estimators are scale invariant. The S-estimator of scale is a family of estimators where this is not the case, hence the name where S stands for scale [52]. They were proposed on the basis of the maximum likelihood estimators (M-estimators) [53]. For the M-estimators, we minimize $\sum_j \rho(r_j)$, where $\rho(t)$ is a symmetric nonconstant function with a unique minimum at zero and is nondecreasing with respect to $|t|$. LS is an M-estimator with $\rho = r^2$. For the S-estimator, we additionally require that ρ is continuously differentiable, $\rho(0) = 0$, and that there exists a constant $k > 0$ such that ρ is strictly increasing in $[0,k]$ and constant in $[k,\infty[$.

A commonly used ρ-function that fulfills the S-estimator requirements is the Tukey bisquare [49], which is defined as

$$\rho_{bis}(t) = \min\{1 - (1 - t^2)^3, 1\}. \tag{20}$$

The saturation with respect to t makes the estimator robust if the residuals are properly scaled, which is done by dividing with an estimate for the scale $\hat{\sigma}$. The loss function for an S-estimator can therefore be written as

$$L = \frac{1}{N}\sum_j \rho\left(\frac{r_j}{\hat{\sigma}}\right), \tag{21}$$

for the scale estimate $\hat{\sigma}$ that is found as the solution of

$$\frac{1}{N}\sum_j \rho\left(\frac{r_j}{\hat{\sigma}}\right) = \delta, \tag{22}$$

where δ determines the breakdown point of $\hat{\sigma}$. If there are more than one $\hat{\sigma}$ that solves this equation, then we take the smallest of them [52]. A maximum breakdown point is obtained with $\delta \approx 0.5$, with a correction that depends on the number of fitting parameters compared to the size of the training dataset [49].

With S-estimators, it is possible to obtain either a high breakdown point or a high efficiency, but not simultaneously both. The S-estimator, however, provides a good starting point for the MM-estimator, which can.

b. The MM-estimator. To simultaneously achieve a high breakdown point and a high normal efficiency, we use the MM-estimator where two M-estimators are used in serial [51]. The first M-estimator ρ_0 is chosen to have a high breakdown point and the second ρ_1 is chosen to have a high efficiency. However, we constrain the second M-estimator ρ_1 such that the breakdown point of ρ_0 is retained by using the scale of ρ_0. The high efficiency of the second M-estimator is obtained by introducing different normalization constants for each M-estimator, given as $\rho_0(r) = \rho(\frac{r}{c_0\hat{\sigma}})$ and $\rho_1(r) = \rho(\frac{r}{c_1\hat{\sigma}})$. To ensure a higher efficiency of ρ_1 compared to ρ_0, we need $\rho_1 \leqslant \rho_0$ and therefore $c_1 \geqslant c_0$ [49]. We will use the S-estimator Tukey bisquare ρ_{bis} for ρ in the MM-estimator, and find the shared robust scale by solving Eq. (22).

c. Weighted normal equations. To solve the RR-LS-type cost function with an S-estimator ρ-function, we can use the iterative reweighting least squares (IRWLS) method [49]. We first note that the solution of the RR-LS cost function can be found in a closed form, given as $\mathbf{a} = (\mathbf{X}^{\mathbf{T}}\mathbf{X} + \mathbf{I}\omega^2)^{-1}\mathbf{X}^T\mathbf{y}$ [44]. For the S-estimator loss function, we can create a similar solution by introducing a weight on each system in the training data [49,54]. These weights are calculated as

$$t_j = \frac{r_j}{\hat{\sigma}}, \quad w_j(t_j) = \frac{\rho'(t_j)}{2t_j},$$
$$\mathbf{w} = (w_1,\dots,w_N), \quad \mathbf{W} = \mathrm{diag}(\mathbf{w}), \tag{23}$$

and are used to scale the design matrix and target vector by $\mathbf{X} \rightarrow \mathbf{WX}$ and $\mathbf{y} \rightarrow \mathbf{Wy}$. Differentiating the cost after the transformations with respect to the coefficient vector $\mathbf{a}$ and setting it equal to zero yields the solution

$$\mathbf{a} = (\mathbf{X}^T\mathbf{WX} + \omega^2\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Wy}, \tag{24}$$

which is a weighted version of the solution to the RR-LS cost function. Since ρ and $W(t)$ are decreasing functions of $|t|$, observations with large residuals relative to the scale $\hat{\sigma}$ will be weighted down through $\mathbf{W}$.

The weighted normal equation is solved through an iterative procedure. Step 0: We initialize with a guess for a robust solution coefficient vector. Step 1: For the coefficient vector, we determine the IRWLS weights according to Eq. (23). Step 2: We solve Eq. (24) and find a new coefficients vector. Step 3: We check for convergence in the weights and terminate if the procedure fulfills our convergence criteria or, if not, jump back to step 1 [55]. For the S-estimator, we also determine the scale estimate by solving Eq. (22) in step 1, whereas the scale is fixed in ρ_1 of the MM-estimator.

d. The Hessian. The Hessian for the cost function with IRWLS weights is given as

$$\hat{\mathbf{H}} = \mathbf{X}(\mathbf{X}^T\mathbf{WX} + \omega^2\mathbf{I})^{-1}\mathbf{X}^{\mathbf{T}}\mathbf{W}, \tag{25}$$

and the corresponding number of effective parameters is found as the trace of this Hessian, hence $\hat{N}_{eff} = \mathrm{Tr}(\hat{\mathbf{H}})$ [45,54].

e. Calculation procedure. We base our implementation of the MM-estimator in our cost function on Ref. [54], which provides a procedure with corrections for integrating the MM-estimator loss function in a ridge-regression-type cost function. To conform with the fitting compromise, we introduce a bias term μ for each dataset, which we calculate the residuals about, such that $r_j = y_j - \mathbf{x}_j\mathbf{a} + \hat{\mu}_i$ for the ith dataset. For each dataset, we estimate the location parameter $\hat{\mu}_i$ and the scale estimate $\hat{\sigma}_i$ simultaneously, using Huber's second method [56].

As the first step for making the MM-estimator, we need to find the S-estimator of scale for each dataset. The IRWLS procedure for solving the S-estimator loss function, however, has to be initiated with a good robust guess $\mathbf{a}_{init}$ for the coefficient vector, i.e., a guess with a high breakdown point. To this end, we take $N_{init} = 200$ regular random bootstrap samples with replacement from the training data and solve the regular RR-LS cost function for each sample [57]. The regularization strength has been determined before we initiate the MM-estimator procedure, as we will describe later. If we have outliers in the training data, then some of the bootstrap samples should omit a part of or all of these outliers, and result in sensible models. These starting guesses might not be robust enough if the datasets were to be highly contaminated with large outliers, but we do not expect that to be the case for our fitting problem. For each guess, we calculate the S-estimator scales $\hat{\boldsymbol{\sigma}}$ for the Tukey bisquare ρ-function through Eq. (22), where for the ith training dataset we use $\delta_i = 0.5[1 - \hat{N}_{eff}/\mathrm{Tr}(\mathbf{W}_i)]$ and normalize with $c_{i,0} = 7.8464 - 34.6565 \cdot \delta_i + 75.2573 \cdot \delta_i^2 - 62.5880 \cdot \delta_i^3$ [58]. This would, for example, yield $c_0 = 1.51$ when $\delta = 0.5$. From the N_{init} solutions, we take the $N_{keep} = 25$ with the lowest weighted geometric mean of the estimated scales for the training datasets.

To find the scale estimate for the MM-estimator, we introduce an S-estimator loss function, which we solve using the IRWLS method for the N_{keep} initial guesses. We label the S-estimator loss function by SE, and it takes the form

$$L_{i,SE}(\mathbf{a}) = \hat{\sigma}_i^2 \sum_j \rho_{bis}\left[\frac{r_j(\mathbf{a})}{c_{i,0}\hat{\sigma}_i}\right], \tag{26}$$

where the $\hat{\sigma}_i^2$ in the front of the summation is a normalization factor introduced to make the loss function coincide with the LS loss function when $\rho(t) = t^2$. We update $\hat{\sigma}_i$ and $c_{i,0}$ in each iteration of the IRWLS procedure according to the expressions in the above paragraph. In the cost function, we scale ω to match RR-LS with $\omega_{SE}^2 = \omega^2/3.43$ [59]. We apply the same scaling when using the MM-estimator loss functions. To achieve consistent results, we converge the IRWLS procedures to a fairly low tolerance of 5‰ on the average change and 10% for the maximum change of the IRWLS weights [60].

From the N_{keep} converged IRWLS solutions, we pick the solution with the lowest weighted geometric mean of estimated scales, which we call $\hat{\mathbf{a}}_{SE}$ with $\hat{\boldsymbol{\sigma}}_{SE}$. We use $\hat{\mathbf{a}}_{SE}$ as the initial solution for the MM-estimator, while the scales for the second step of the MM-estimator are found with the correction to $\hat{\boldsymbol{\sigma}}_{SE}$ given as

$$\hat{\sigma}_{i,MM} = \frac{\hat{\sigma}_{i,SE}}{1 - [k_1 + k_2/\mathrm{Tr}(\mathbf{W}_{i,SE})] \cdot N_{eff}/\mathrm{Tr}(\mathbf{W}_{i,SE})}, \tag{27}$$

where $k_1 = 1.29$, $k_2 = -6.02$, and $\mathbf{W}_{i,SE}$ is the part of the converged SE IRWLS weights $\mathbf{W}_{SE}$ that belongs to the ith dataset [61]. The MM-estimator loss function is then given as

$$L_{i,MM}(\mathbf{a}) = \hat{\sigma}_{i,MM}^2 \sum_j \rho_{bis}\left[\frac{r_j(\boldsymbol{a})}{c_1\hat{\sigma}_{i,MM}}\right], \tag{28}$$

where we choose $c_1 = 3.44$ as in Ref. [54] to provide a nominal efficiency of 85% as a higher efficiency has been found to introduce a bias. We solve the cost function with the MM-estimator loss function using the IRWLS method similar to that for the SE loss function, but where the scales are now kept fixed.

f. Convergence of the regularization strength. For convergence of the IRWLS weights with respect to the regularization strength, we use the following procedure. Step 0: We initialize with $\mathbf{W} = \mathbf{I}$. Step 1: We find the optimal regularization strength by minimizing the EPE, where we have scaled all data points with $\mathbf{W}$ in Eq. (10) with the LS loss function. Step 2: For the optimal regularization strength, we solve with the MM-estimator loss function, which gives us the IRWLS weights that will make the LS loss function solve as the MM-estimator. If the procedure has not converged, then we go back to step 1 and use the new weights to find the optimal regularization strength again. We used a convergence tolerance of 10% on the maximum residual change and 1% on average residual change between the iterations. The procedure converged to fixed final solution vector $\hat{\mathbf{a}}_{MM}$ and scales $\hat{\boldsymbol{\sigma}}_{MM}$ in 5–10 iterations [62].

D. Bayesian error estimation ensemble

As with previous BEEF functionals [7–9], we propose an ensemble of functionals to be used for error estimation of DFT predictions. We use a probability distribution P to generate the ensemble of fluctuations $\delta\mathbf{a}$ around $\mathbf{a}_0$, defined as $P \propto \exp[-\tilde{K}(\mathbf{a})/\tau]$, where we have to set a "temperature" τ. Practically, we choose τ such that the ensemble reproduces the weighted geometric mean of the observed error for the fitted datasets. To do this, we note that the unscaled, i.e., $\tau = 1$, average ensemble error for all fitted data points can be found by $\mathbf{y}_{BEE} = \sqrt{\mathrm{Tr}(\mathbf{X}\hat{\mathbf{H}}\mathbf{X}^T)}$. We label the root-mean-square error for the observed error by $\mathrm{RMSE}_{observed}$ and for the Bayesian error estimation by RMSE_{BEE}. The temperature is then given as

$$\tau = \left[\frac{\left(\prod_i \mathrm{RMSE}_{observed,i}^{\mathcal{W}_i}\right)^{1/\sum \mathcal{W}_i}}{\left(\prod_i \mathrm{RMSE}_{BEE,i}^{\mathcal{W}_i}\right)^{1/\sum \mathcal{W}_i}}\right]^2, \tag{29}$$

where the index i is for the ith datasets. We define a scaled inverse Hessian $\boldsymbol{\Omega}$ that we use to create the ensemble functions as

$$\boldsymbol{\Omega} = \tau\hat{\mathbf{H}}^{-1}, \tag{30}$$

which has the eigenvalue decomposition $\boldsymbol{\Omega}\mathbf{V} = \mathrm{diag}(\mathbf{u})\mathbf{V}$, with the eigenvalues $\mathbf{u}$ on the diagonal of a square matrix with zero nondiagonal elements and eigenvectors in the rows of the matrix $\mathbf{V}$. The ensemble functionals can then be generated as

$$\delta\mathbf{a}_k = \mathbf{V} \cdot \mathrm{diag}(\sqrt{\mathbf{u}}) \cdot \mathbf{r_k}, \tag{31}$$

where $\mathbf{r_k}$ is a N_p long random vector of normal distributed numbers (variance 1 and mean 0). For a single DFT energy, we can also find the average error estimate directly as $\sqrt{\mathbf{x}\mathbf{\Omega}^{-1}\mathbf{x}^T}$, where $\mathbf{x}$ is a vector with the basis function energies for the converged calculation.

III. OPTIMIZING FUNCTIONAL APPROXIMATION

In the following, we show the performance sensitivity to the most important parameters in the fitting methodology. We set the weights $\mathcal{W}_i$ for the datasets to the following: 2 for CE27 and RE42, 1 for Sol54Ec and Sol58LC, and $\frac{1}{5}$ for each S22 × 5 subset. The weights on S22 × 5 hence add up to 1, which puts the full dataset on the level with Sol54Ec and Sol58LC. The choices follow those made for BEEF-vdW in Ref. [9], but with a higher weight on S22 × 5 because we want to use the enlarged model space to improve prediction power on dispersion dominated systems.

For the regularization, we use $\lambda_{\alpha/s} = 10$ and $\lambda_{c/x} = \lambda_I = 10^{-4}$ as those values seem to produce the lowest EPE. For origo of the coefficients $\mathbf{a}_p$, we use $F_x(s,\alpha) = 1, \alpha_{LDA} = 0.5$, $\alpha_{sl} = 0.5$, and $\alpha_{nl} = 1$. We generate 500 hierarchical bootstrap samples and reuse them for all regularization strengths, so that the EPEs at different regularization strengths are comparable [63].

A. Convergence of geometries and electronic densities

The XC functional can be optimized through a linear fit because the total energy depends linearly on the fitting parameters. However, this only holds if self-consistency of the electronic density can be neglected. To address this issue, we make two iterations of optimizing the geometries and densities with subsequent functional fittings. In the first iteration, we create the basis function energies for the fit using geometries and densities from converged calculations with PBEsol [36]. We refer to the fit made on the PBEsol densities as release candidate 1 (RC1) [64]. We then reoptimize all systems in the training datasets with the RC1 functional and calculated new energies for each basis function at the converged RC1 geometries and densities. For the final fit, we use the converged densities to RC1. We refer to this fit as release candidate 2 (RC2) or mBEEF-vdW. We reoptimize all systems in the training datasets again to the RC2 functional for the benchmark of mBEEF-vdW. To assess self-consistency, we compare the root-mean-square difference for each dataset between the prediction on RC1 densities and the self-consistent result: ∼10% for Sol54Ec, <1% for the geometric mean of S22 × 5, virtually zero for CE27, and 3% for RE42. We do not have non-self-consistent predictions for the lattice parameters for RC2 to compare with unfortunately.

B. Regularization strength

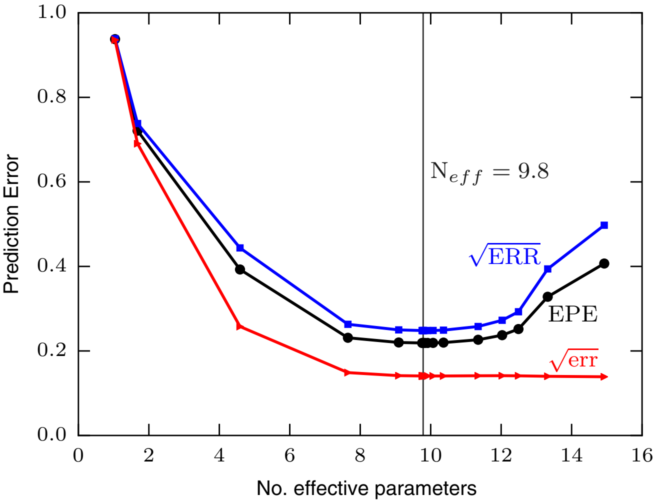

FIG. 1. Prediction error as a function of regularization strength. We plot the estimated prediction error (EPE), the bootstrap error estimation (ERR), and the training error (err) of the hierarchical 0.632 bootstrap method with weighted geometric mean. We display the square root of the ERR and err to make them of the same units as the EPE.

The effect of the regularization strength on the EPE is shown in Fig. 1, where we find a minimum EPE at 9.8 effective parameters. The err does not change much from around 8 effective parameters, indicating that overfitting will not improve performance much. The ERR increases slightly around the minimum and sharply upwards at around 13 effective parameters, indicating overfitting if we were to use that many effective parameters. The high EPE, ERR, and err at few effective parameters shows that the performance is drastically improved from the origo solution vector $\mathbf{a}_p$.

C. Correlation functions

In Table I, we show how different semilocal and nonlocal correlation functionals affect the EPE, ERR, and err. The best result was found with PBEsol and vdW-DF2, which has the lowest err, ERR, and EPE, and also the lowest number of effective parameters. We observe a spread in the EPE of about 20% between the different choices of correlation terms, and vdW-DF2 nonlocal correlation performs better than vdW-DF in all pairings. For vdW-DF2, one might propose that the PBEsol correlation is favorable due to the PBEsol starting geometries and densities. We cannot exclude that the starting geometries and densities can play a role in what correlation performs the best, but the geometries and densities of RC1 are significantly different from those made with PBEsol. The starting choice therefore has a negligible effect.

It is noticed that the following pattern shows for EPE: PBE > vPBE > PBEsol. These correlation functionals only differ by the value of the parameter β: $\beta_{PBE} = 0.0667 > \beta_{vPBE} >$

TABLE I. Optimal EPE, ERR, and err when using different semilocal and nonlocal correlation functionals.

	N_{eff}	ERR	err	EPE
vdW-DF2+vPBE	10.0	256	144	222
vdW-DF2+revTPSS	12.5	272	149	235
vdW-DF2+PBEsol	9.7	249	139	215
vdW-DF2+PBE	10.1	258	146	223
vdW-DF+vPBE	11.7	298	156	255
vdW-DF+revTPSS	12.7	315	151	266
vdW-DF+PBEsol	11.9	316	152	268
vdW-DF+PBE	11.8	300	156	257

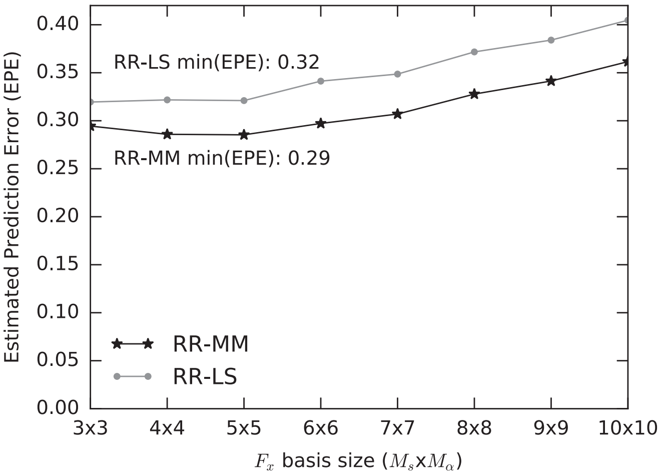

FIG. 2. EPE as a function of the number of basis functions in the expansion of the exchange enhancement function for the robust MM and the LS loss functions. The optimal number of basis functions are $M_s, M_\alpha = 5$ for the MM loss function, with a 10% improvement over the least-squares loss function minimum at $M_s, M_\alpha = 3$. There are only small variations between training error for the two loss functions, but with the MM loss function slightly lower.

$\beta_{PBEsol} = 0.046$. Therefore, it seems that a smaller value of β might be able to increase the accuracy of the functional further. It might even be fruitful to go all the way to the low-density limit, $\beta = 0.038$, which has been used in the correlation term of the functional SG4 [65].

D. Exchange basis size

Figure 2 shows how the number of basis functions for the exchange expansion affects the EPE. The optimum number of exchange parameters for the robust MM loss function is $M_s, M_\alpha = 5$, while for regular LS loss function it is $M_s, M_\alpha = 3$. Comparing the lowest EPE of the MM and the LS loss functions, we observe that using the MM estimator leads to a reduction in EPE of 10%.

In Fig. 3, we show the training error and find that the spread is smaller between the MM and LS loss functions, where MM yields the lowest training error, but only by a small amount [66].

We note that we used the full fitting procedure to determine the optimal fits and the EPE for each basis size, and that the EPE of MM and LS are directly comparable as only the coefficient vectors are different in their respective estimations.

The MM method generally results in more effective parameters than LS, and this could explain why it has a lower err. With the LS loss function, the regularization strength is increased to avoid overfitting as the only defense against outliers. The MM estimator will weigh these outliers down and will therefore not be affected by them. However, the difference between the LS and MM loss functions is much larger for the EPE, where we test the transferability of the fit to data outside the training data. We reason that this is because the outliers in the bootstrap sample do not provide a description of the underlying model, which would be transferable to the training data systems excluded in the sample. We therefore select more transferable models by weighing down these outliers.

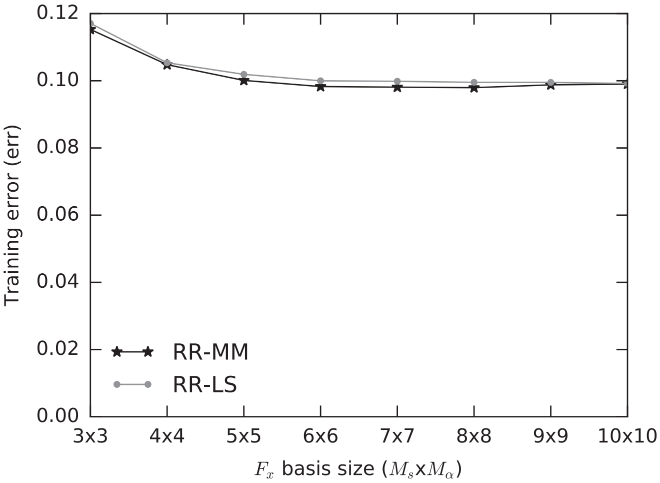

FIG. 3. Training error (err) as a function of the number of exchange enhancement basis functions for the robust MM and the least-squares (LS) loss functions. The MM loss function performs better for all choices of basis sizes, but only by a small amount.

Outliers in the datasets

The IRWLS weight vector **w** tells us which systems the MM loss function identifies as outliers. We can therefore compare how affected each dataset is by their normalized sum of **w** [67]. There are almost no outliers in the CE27 and RE42 datasets with $\sum \mathbf{w}/N_i \simeq 0.95$ for both. For the two solids datasets Sol54Ec and Sol58LC, we identify a larger number of outliers with $\sum \mathbf{w}/N_i$ equal to 0.69 and 0.65, respectively. The S22 × 5 sub datasets also have a similar proportion of outliers, with $\sum \mathbf{w}/N_i$ ranging from 0.62 to 0.70. The outliers in the S22 × 5 subdatasets are shared to a large extend between the different binding lengths.

We can generally divide the cause of an outlier into three categories: (1) Bad reference data due to inaccurate or error prone experiments or reference computations. (2) Unfit model systems for experiments due to, for example, not taking into account dislocations, defects, or impurities in crystal structure. (3) Model space deficiencies due to, for example, self-interaction error, lack of spin-orbit coupling, convergence issues, or too crude atomic core descriptions.

For the solids and chemisorption datasets, the outliers could be caused by all three effects. However, for the S22 and RE42 datasets, the reference values are high-quality coupled cluster with single, double, and partially triple excitation [CCSD(T)] data, which we expect are much more accurate than our DFT model. The outliers for these are therefore primarily due to a limiting DFT model. For the latter, we would need to compare to more sophisticated methods to determine what is the primary cause of the outliers. Such an investigation is, however, beyond the scope of this study.

IV. THE mBEEF-vdW FUNCTIONAL

For fitting the mBEEF-vdW functional, we used $M_s = M_\alpha = 5$, hence 25 exchange enhancement basis functions. We therefore have a total of 28 fitting coefficients when we include the three correlation coefficients. The correlation basis functions are LDA, PBEsol, and the nonlocal part of

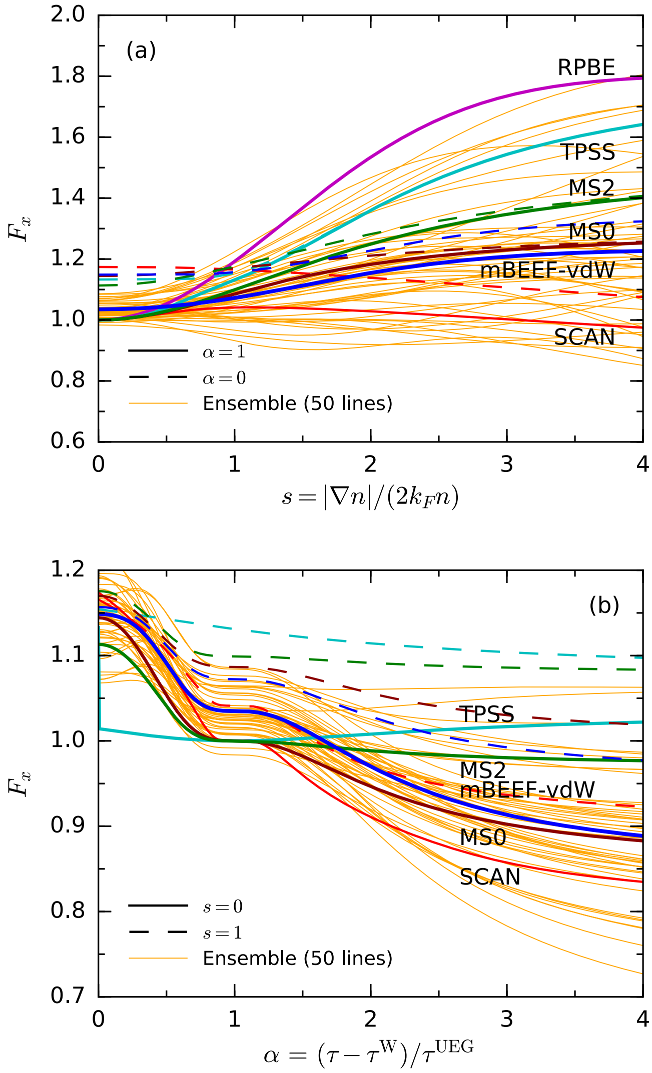

FIG. 4. The exchange enhancement factor of mBEEF-vdW and 50 representative mBEEF-vdW ensemble functionals with reference to a couple of popular semilocal exchange-correlation functionals. We present cuts in the two-dimensional space of the reduced density gradient, s, and the reduced kinetic energy density, α. The cross section between the two cuts for, namely, $\alpha = 1$ and $s = 0$ corresponds to the homogeneous electron gas. SCAN is found in [21].

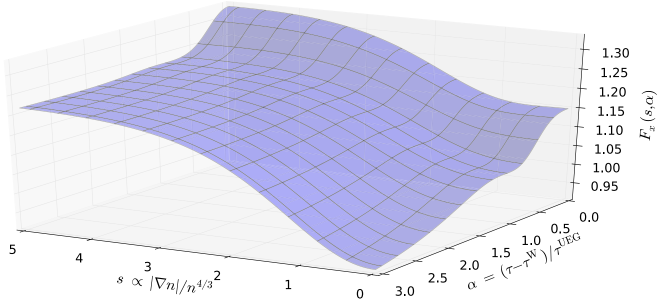

FIG. 5. The exchange enhancement factor of mBEEF-vdW as a function of s and α.

vdW-DF2. The parameters of the mBEEF-vdW functional can be found in the Supplemental Material [68], but with illustrations following here.

Figure 4 shows the exchange enhancement factor of the mBEEF-vdW functional with a representative BEE ensemble along common representative planes in the s,α space. The functional parametrization cannot be defined as a composition of a function of α and s independently, unlike MGGA functionals such as MS0 and MS2. In Fig. 5, we therefore include a three-dimensional (3D) visualization and observe that the functional varies smoothly with s and α, which was expected due to the restricted basis of the exchange enhancement factor.

To complement the visual inspection, we here provide some relevant limits of the exchange enhancement factor: $F_x(s = 0,\alpha = 1) = 1.035$, $F_x(s = 0,\alpha = 0) = 1.149$, $F_x(s \to \infty,\alpha = 1) = 1.194$, and $F_x(s \to \infty,\alpha = 0) = 1.286$ (maximum value). We observe that mBEEF-vdW breaks the LDA limit ($s = 0$ and $\alpha = 1$), but is close to MS0 in the limits $\alpha = 0$ and $\alpha \to \infty$ for $s = 0$ and for $s \to \infty,\alpha = 1$. We note that the MS0 functional form was used to define the basis functions transformation of the alpha space, which could affect these limits. The recent MGGA functional, namely the strongly constrained and appropriately normed (SCAN) functional [21], deviates from mBEEF-vdW in all the previous discussed limits.

For the correlation parameters of mBEEF-vdW, we find the optimal coefficient: 0.41 ± 0.38 for LDA, 0.36 ± 0.40 for PBEsol, and 0.89 ± 0.31 for vdW-DF2 nonlocal correlation, where the ± intervals are the standard deviations of the BEE ensemble for each coefficient. We note that the mBEEF-vdW functional therefore does not fulfill the LDA limit of the correlation functional, in contrast to the BEEF-vdW functional [9]. The nonlocal coefficient on vdW-DF2 is about 90% of the full value, which also is a departure from what is theoretically justified for the LDA limit. The vdW-DF2 functional is, however, also acting as a short-range functional, and if the nonlocal coefficient had been 1, the mBEEF-vdW would have too much semilocal correlation as it also includes short-range correlation from PBEsol.

Figure 6 shows the enhancement factors of mBEEF-vdW and the previous two BEEF functionals: BEEF-vdW and mBEEF. We observe that all three functionals break the LDA limit for the exchange enhancement ($\alpha = 1$ and $s = 0$) by a nearly identical amount, even when they are the results of somewhat different fitting procedures, different training data, and different model complexities. We also note that mBEEF-vdW rises slower from the homogeneous electron gas limit than mBEEF and BEEF-vdW, and mBEEF-vdW is also smoother than mBEEF and BEEF-vdW.

BEE estimates. In Table II, we compare the root-mean-square error (RMSE) of mBEEF-vdW with the Bayesian error ensemble estimate for each training dataset. The BEE estimate is found as the root-mean-square sum of the BEE estimates for each system in the dataset. For CE27a, RE42, and S22x5, we base this evaluation on the self-consistent mBEEF-vdW results and BEE predictions, whereas for the two solids datasets Sol54Ec and Sol58LC the results are non-self-consistent predictions. The difference between the self-consistent and non-self-consistent RMSE and BEE estimate is less than

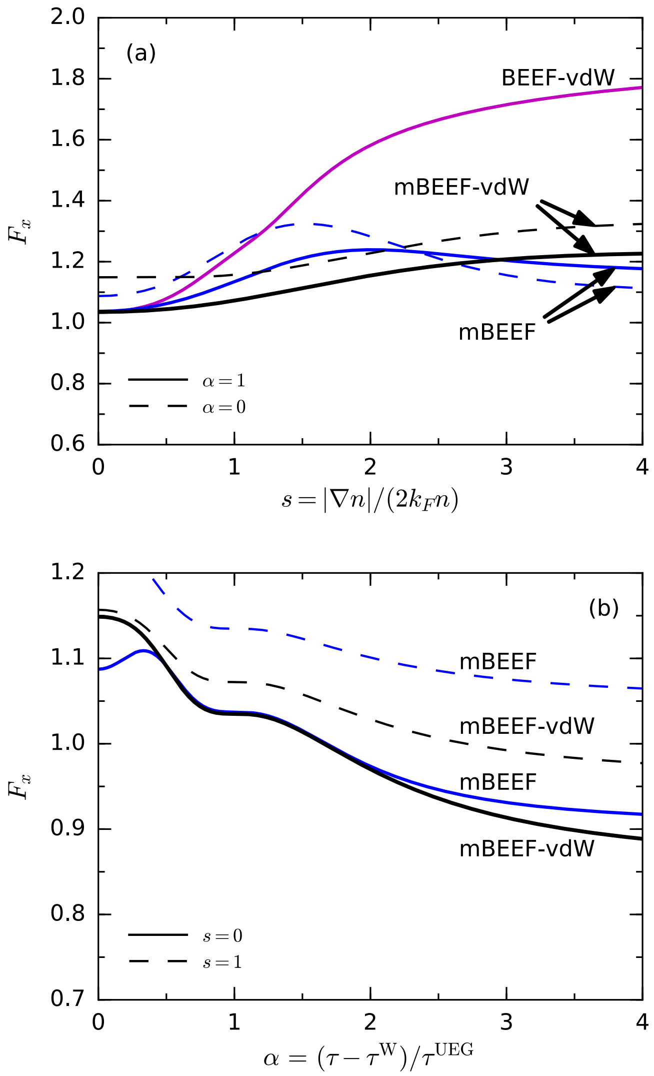

FIG. 6. The exchange enhancement factors of mBEEF-vdW, mBEEF, and BEEF-vdW. We present cuts in the two-dimensional space of the reduced density gradient, s, and the reduced kinetic energy density, α.

10% for CE27a, S22 × 5, and RE42, and non-self-consistent evaluation for the solids datasets should therefore suffice [69].

The estimated error is within a factor of two of the real error for all datasets. For S22 × 5, we find that the error estimation for mBEEF-vdW is much more accurate than with BEEF-vdW, where a threefold difference was found between its BEE estimates and the actual error [70].

TABLE II. Comparing RMSE of mBEEF-vdW to its BEE error estimate.

	RMSE	BEE	RMSE/BEE
CE27a (eV)	0.25	0.42	0.59
RE42 (eV)	0.45	0.38	1.2
S22 × 5[a] (meV)	14.5	17.6	0.82
Sol54Ecoh (eV)	0.40	0.26	1.6
Sol58LC-dEc (eV/Å)	23.2	16.6	1.4

[a]Geometric mean over subsets.

V. RESULTS

We benchmark mBEEF-vdW against popular lower-rung XC functionals on the training datasets and two relevant surface science test datasets. For the benchmarked datasets, we illustrate the performance compromises with bivariate analyses of several interesting dataset pairs. Lastly, we present the test case of graphene on a nickel(111) surface to learn how mBEEF-vdW deals with the interplay between chemisorption and physisorption. This problem also provides an illustration of how to use mBEEF-vdW's Bayesian error estimating ensemble.

In the Supplemental Material [68], we additionally provide the results for finding the correct binding site for CO on late transition metals, similarly to what was presented in Ref. [7].

A. Benchmark of mBEEF-vdW

Figure 7 shows a benchmark of the mBEEF-vdW functional with popular or recent GGA, MGGA, and vdW-DF density functionals on the mBEEF-vdW training datasets. The following functionals are listed with citations. GGA type: PBEsol [36], PBE [35], RPBE [71]; MGGA type: TPSS [31], revTPSS [37], oTPSS [72], MS0 [33], MS2 [29]; GGA-vdW type: vdW-DF [28], vdW-DF2 [27], optB88-vdW [73], optPBE-vdW [73], C09-vdW [74]. The first panel ranks the functionals according to the geometric mean (GM) of the root-mean-square error (RMSE) for the five datasets' statistics relative to mBEEF-vdW. The other panels show the RMSE for the considered functionals on each training dataset. For S22 × 5, we show the geometric mean of RMSE over its five subsets.

The mBEEF-vdW functional is the highest ranked functional, as we expected because it is trained on these datasets and has the most advanced model space. The other highly ranked functionals are the two previous BEEF family functionals mBEEF and BEEF-vdW, the optimized vdW-DF functionals optB88-vdW and optPBE-vdW, and the MS family functionals MS2 and MS0.

The mBEEF-vdW's RMSE for CE27a, Sol58LC, and S22 × 5 is among the lowest for all functionals tested, while it has a relative modest accuracy on Sol54Ec and RE42 compared to the other functionals. For the absorption energies of CE27a, mBEEF-vdW has a matching performance to the BEEF family functionals and RPBE. For the lattice constants of Sol58LC, mBEEF-vdW has the lowest prediction error of all functionals tested, even surpassing PBEsol and the MS functionals. For the dispersion systems in S22 × 5, the performance of mBEEF-vdW is at a level compared to the optimized vdW functionals of optB88-vdW and optPBE-vdW.

Figure 8 shows a benchmark on the SE30 datasets for surface energies of 30 systems and BM32 for bulk moduli of 32 systems as described in Ref. [7]. We note that optB88-vdW is not present for SE30 because we were not able to converge its electronic density. We observe that mBEEF-vdW has the lowest RMSE on the SE30 dataset of all the functionals tested and performs moderately well on the BM32 dataset.

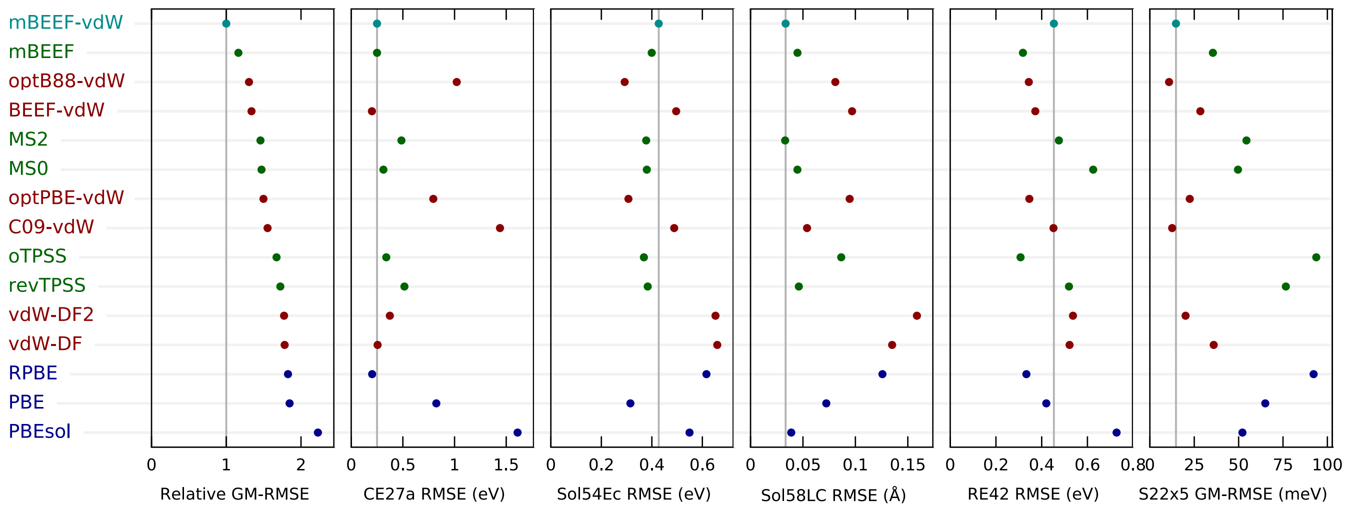

FIG. 7. Benchmark of mBEEF-vdW against popular or recent GGA (blue), MGGA (green), and vdW-DF (red) density functionals in terms of root-mean-square error (RMSE) on the training datasets. The first panel ranks the tested density functionals according to the geometric mean of the five datasets. All results have been obtained self-consistently.

The results on these datasets indicate that the performance of mBEEF-vdW is transferable to systems that it was not trained on.

Bivariate analysis

Figure 9 shows chemisorption energies (CE27a) versus surface energies (SE30), and we observe that mBEEF-vdW is able to achieve a high accuracy of both properties simultaneously, unlike most other tested functionals. The functional rungs clearly stand out as reported in Ref. [7], and the expanded model space of mBEEF-vdW is therefore a likely explanation for its higher accuracy.

Figure 10 shows chemisorption energies (CE27a) versus dispersion energies (S22 × 5), and we observe that the mBEEF-vdW is able to achieve a high relative accuracy on both. The GGA-vdW density functionals generally improve over the GGA functionals along the dispersion axis, but still make a trade-off between the accuracy on these two datasets.

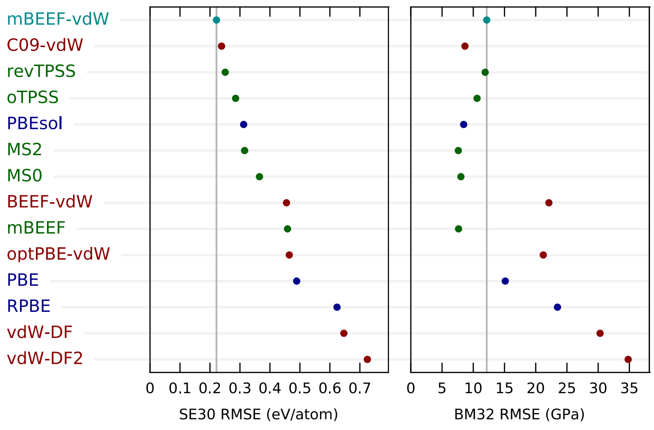

FIG. 8. Benchmark on the datasets SE30 with 30 surface energies and BM32 with bulk modulus of 32 systems. For both datasets, we show the root-mean-square error. The functionals are ranked by the performance on SE30.

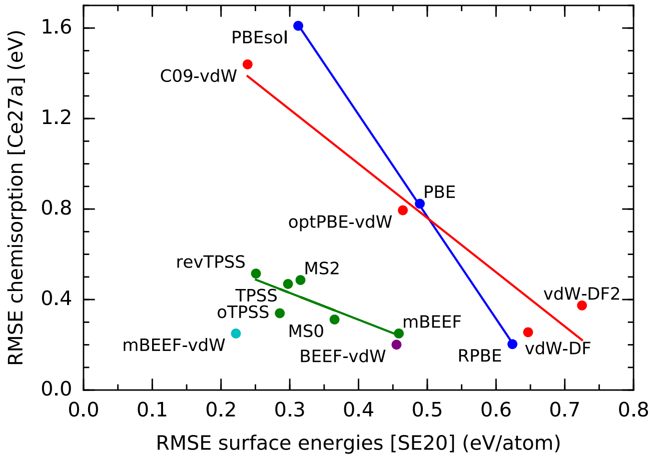

FIG. 9. Bivariate analysis of chemisorption and surface energies, given by the datasets CE27a and SE30. The lines are linear first-order fits to the functional of the same color.

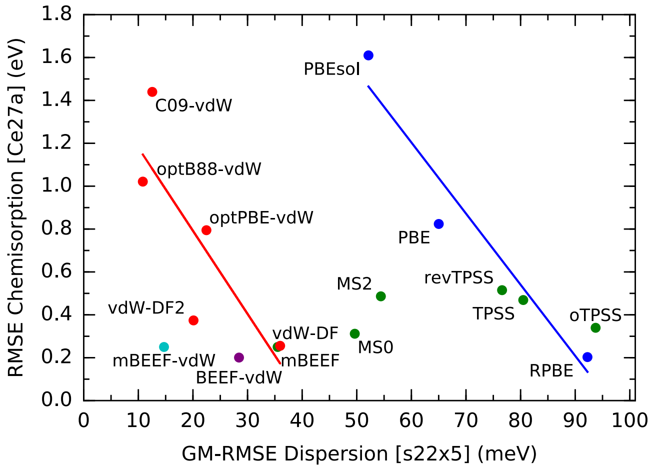

FIG. 10. Bivariate analysis of chemisorption and dispersion, given by the datasets CE27a and S22 × 5.

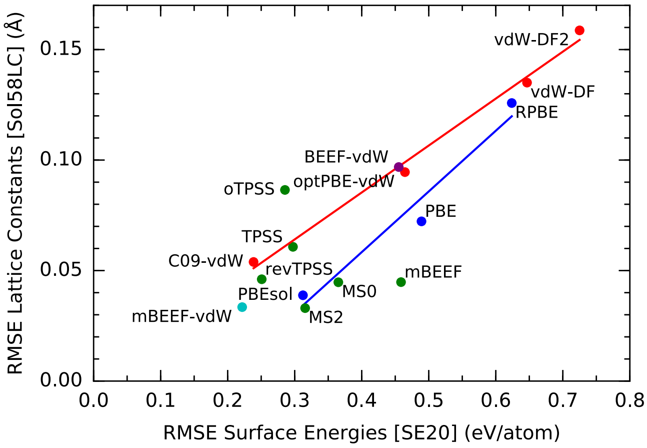

FIG. 11. Bivariate analysis of lattice constants and surface energies, given by the datasets Sol58LC and SE30.

vdW-DF2 performs reasonably well on both properties, but with a lower accuracy on both compared to mBEEF-vdW.

Figure 11 shows lattice constants (Sol58LC) versus surface energies (SE30), and we can clearly see that there is a high correlation between the two properties. In Fig. 12, we can similarly observe a high correlation between lattice constants (Sol58LC) and bulk moduli (BM32). The high correlation between the Sol58LC and SE30 datasets suggests that mBEEF-vdW has a high accuracy on the surface energies dataset because it has a high accuracy on lattice constants.

In the Supplemental Material [68], we additionally show the following bivariate plots: BM32 and RE42, S22 × 5 and RE42, Sol58LC and RE42, S22 × 5 and SE30, CE27a and Sol54Ec, and CE27a and RE42. With the last two bivariate plots, we can attribute mBEEF-vdW's modest description of cohesive energies and reaction energies to the model compromise. The benchmarked functionals form a frontier on these properties, but with a few functionals that are inferior on both properties including mBEEF-vdW. We attribute the compromises the fit has to do to the other material properties in the training datasets.

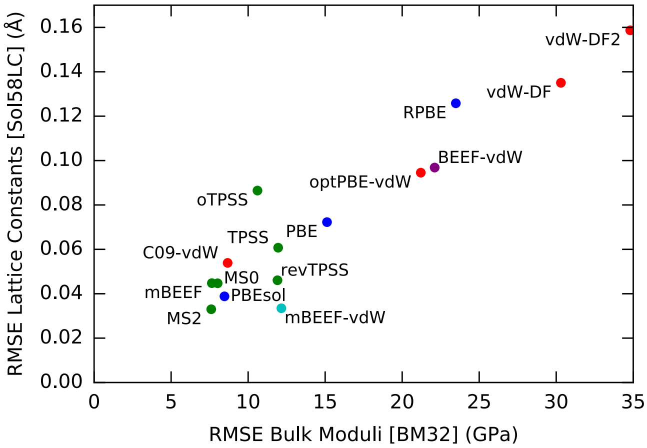

FIG. 12. Bivariate analysis of lattice constants and bulk moduli, given by the datasets Sol58LC and BM32.

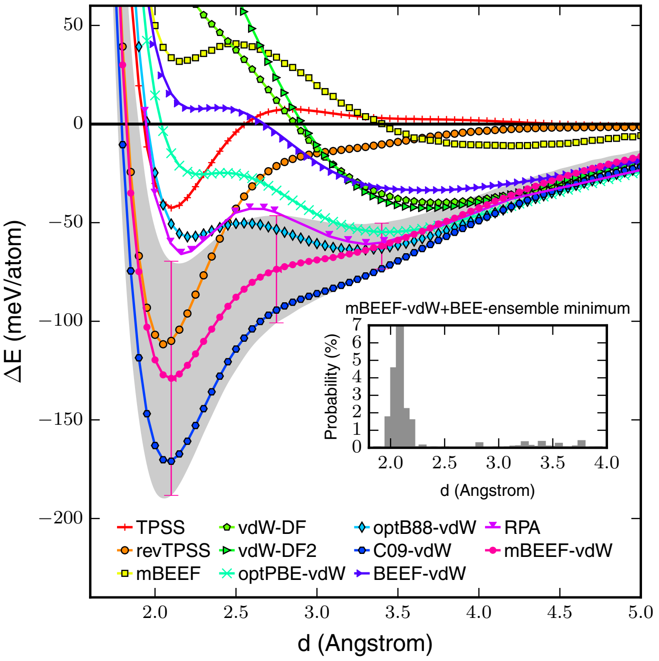

FIG. 13. Potential-energy curves for graphene adsorption on Ni(111) surface. The gray area indicates the region spanned by the estimated standard deviations along the mBEEF-vdW's potential-energy curve. Inset shows the distribution of predicted binding length for 10.000 mBEEF-vdW BEE ensemble functionals. Random phase approximation data from Ref. [75].

B. Graphene adsorption on Ni(111)

Figure 13 show the potential-energy curve for graphene on the nickel (111) surface, which we can use to qualify how well the mBEEF-vdW balances covalent and vdW forces. This system has been investigated in numerous computational studies [76–84] and it is experimentally known that graphene forms a (1 × 1) overlay on the Ni(111) surface with a graphene–metal distance of $d = 2.1$ Å [85].

We find a graphene–metal distance for mBEEF-vdW of 2.10 Å, in agreement with the experimental results. A number of other functionals reproduce the experimental binding length as well, including the MGGA functionals TPSS and revTPSS, and the vdW-DF functional C09-vdW.

Results from the more computationally expensive random phase approximation (RPA) method have indicated the presence of a minimum in the potential-energy curve due to physisorption at $d = 3.0$–3.5 Å. This is also observed with the optimized vdW-DF functionals optB88-vdW and optPBE-vdW, as well as the MGGA functional M06L [75,83,84]. The potential-energy curve for mBEEF-vdW does not include a physisorption minimum, but we can use the BEE ensemble to estimate the likelihood for such a minima for the functionals in the Bayesian error estimating ensemble of mBEEF-vdW. In the inset of Fig. 13, we present the result of such an investigation. Most of the ensemble functionals predict a binding distance around 2.1 Å, but with a heavy tail of about 10% of the ensemble functionals that predicts binding distances longer than 2.5 Å. These longer binding lengths are mostly between 3.2 Å and 3.8 Å, which match the physisorption minimum mentioned earlier.

VI. SUMMARY, DISCUSSION, AND CONCLUSION

We presented the exchange-correlation functional mBEEF-vdW, a Bayesian error estimating functional for applications in particular in heterogeneous catalysis studies. We achieved improvements over previous BEEF functionals by fitting mBEEF-vdW with a parametrization of the MGGA-vdW-type functional space and by improving the fitting procedure. To improve the fitting procedure, we introduced a robust regression loss function, which makes the fit resilient to outliers in the training datasets. We used this loss function in a cost function with weighted geometric mean over loss functions for multiple datasets to make an explicit compromise between different material properties. The model complexity was controlled by a regularization with a nonsmoothness penalty of the exchange enhancement basis functions. To better find the optimal model complexity, we furthermore introduced a generalized bootstrap 0.632 error prediction estimator. The generalization uses a hierarchical bootstrap sampling of the training datasets and the geometric mean over these datasets in its error prediction estimator, thereby making the cross validation more resilient to correlations in the training datasets. The robust MM-estimator loss function resulted in a 10% improvement of the estimated prediction error over the standard least-sum-of-squares loss function.

The mBEEF-vdW functional was trained and benchmarked on datasets of relevance to heterogeneous catalysis. This benchmark showed that the mBEEF-vdW functional is simultaneously one of the most accurate functionals for chemisorption on surfaces, dispersion energies, and lattice constants for the popular density functionals tested in this study. The benchmark included two validation datasets for surface energies and bulk moduli. For surface energies, the mBEEF-vdW was the best performing of all tested functionals and it also had a good performance on the bulk moduli dataset. We lastly tested mBEEF-vdW on the case of graphene adsorbed on the nickel(111) surface, where it correctly predicted the experimental binding length, unlike the previous BEEF functionals and most of the tested XC functionals.

Following the methodology from previous BEE functionals, we provide a functional ensemble to estimate calculation uncertainty due to the exchange-correlation functional approximation. The ensemble for the mBEEF-vdW functional was scaled to reproduce the observed training set errors, with a scaling that takes the robust loss function into account. We found that the RMS sum of the error estimates for the training datasets are all within a multiple of two of the real RMSE. To illustrate the use of the ensemble, we applied it to the graphene on nickel case. Here the BEEF ensemble estimates the likelihood of the existence of a physisorbed binding state to about 10%, which is interesting as several other functionals including RPA found the existence of such a physisorption minimum in the potential-energy curve.

The optimized exchange enhancement factor of mBEEF-vdW was found to slightly break the LDA limit, similar to what was found for previous BEEF family functionals. The exchange enhancement factor of mBEEF-vdW fulfills the tight Lieb-Oxford bound, and it is close to MS0 [33] for the limit of single electron orbitals.

The computational cost of the mBEEF-vdW is higher than GGA functionals because of its use of MGGA exchange and the vdW-DF2 correlation term. However, the computational increase due to the vdW nonlocal correlation part is usually small due to the efficient implementation scheme of Ref. [86], and MGGA exchange similarly does not add much additional computational cost. We have not experienced any additional computational difficulties, including convergence issues, for mBEEF-vdW compared to MGGA functionals and GGA-vdW functionals. We therefore suggest that, with proper implementation, the small added computational cost of mBEEF-vdW is outweighed by its increased accuracy for most purposes.

Our benchmark was restricted to only a subset of available XC functionals. We therefore note that there are a number of other good functionals available to the DFT users. Further benchmarks will determine how mBEEF-vdW compare to these. We also do not benchmark on a number of structural properties, such as bonding lengths and vibration frequencies, which should be taken into account in future studies. The training datasets could be expanded such that these and other material properties are better accounted for.

It has been postulated that the nonlocality of MGGAs makes it possible to describe excitonic effects in semiconductors [87]. However, for this it is required that $|\partial F_x/\partial\alpha|$ is large in the relevant region for solids ($\alpha \approx 1$ and $s < 3$). In Ref. [87], they found that VS98 was able to do this, while TPSS, which has a much smaller α dependence, was not. In Fig. 4, we see that $|\partial F_x/\partial\alpha|$ of mBEEF-vdW is much larger than that of TPSS. One would therefore expect mBEEF-vdW to behave better than TPSS for describing excitonic effects in semiconductors. However, further studies are required in order to know how mBEEF-vdW compares to VS98 with respect to this property.

For semilocal functionals, the XC energy can be evaluated exactly as a sum of the XC functional evaluated on the pseudodensity plus projector augmented wave (PAW) corrections for each atom. However, for a nonlocal functional such as mBEEF-vdW, the PAW correction has not been implemented in GPAW and the results may therefore be sensitive to the details of the PAW dataset such as cutoff radius and choice of pseudocore density—with a PAW correction, this would not be the case. Had a full implementation of nonlocal vdW been used for this work, the resulting mBEEF-vdW functional might have been slightly different.

The mBEEF-vdW is available for GPAW and other DFT codes through the Libxc [88] library in its version 3.0.0 release. We are in the process of implementing the functional in VASP [89–92] and QUANTUM ESPRESSO [93], and we plan to report any differences between the codes.

For training and validation of mBEEF-vdW, we only included a limited number of datasets of high relevance to heterogeneous catalysis, and as such there are a number of material properties that were not covered, which could be of interest to potential users; for example, molecular bond lengths, vibration frequencies, and barrier heights. We suggest that future benchmark should include these. In addition, for future studies, we think it would be beneficial to expand and improve the current training and validation datasets by, for example, using the dispersion dataset S66 × 8 instead of S22 × 5 [94], including more solid-state systems [95], and reevaluate the chemisorption dataset [96]. For developing more accurate BEEF functionals, we also need to address the self-interaction error. This could be done by either introducing (screened) exact exchange or by using a self-interaction correction scheme, such as Hubbard+U or SIC [97,98].

We propose the mBEEF-vdW functional as a well-suited lower-rung XC functional for heterogeneous catalysis studies. Furthermore, we propose that the machine-learning procedure introduced here could lead to more accurate empirical XC functionals in the future.

ACKNOWLEDGMENTS

We acknowledge financial support to SUNCAT from the Chemical Sciences, Geosciences and Biosciences Division of the U.S. Department of Energy' s Office of Basic Energy Science. We also wish to thank the Danish Center for Scientific Computing for computational resources.

[1] W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965).
[2] W. Kohn, A. D. Becke, and R. G. Parr, J. Phys. Chem. **100**, 12974 (1996).
[3] E. A. Carter, Science **321**, 800 (2008).
[4] G. E. Scuseria and V. N. Staroverov, in *Theory and Applications of Computational Chemistry: The First Forty Years*, edited by C. Dykstra *et al.* (Elsevier, Amsterdam, 2005), p. 669.
[5] Y. Zhao and D. G. Truhlar, Theor. Chem. Acc. **120**, 215 (2007).
[6] N. Mardirossian and M. Head-Gordon, J. Chem. Phys. **142**, 074111 (2015).
[7] J. Wellendorff, K. T. Lundgaard, K. W. Jacobsen, and T. Bligaard, J. Chem. Phys. **140**, 144107 (2014).
[8] J. J. Mortensen, K. Kaasbjerg, S. L. Frederiksen, J. K. Nørskov, J. P. Sethna, and K. W. Jacobsen, Phys. Rev. Lett. **95**, 216401 (2005).
[9] J. Wellendorff, K. T. Lundgaard, A. Møgelhøj, V. Petzold, D. D. Landis, J. K. Nørskov, T. Bligaard, and K. W. Jacobsen, Phys. Rev. B **85**, 235149 (2012).
[10] A. J. Medford, J. Wellendorff, A. Vojvodic, F. Studt, F. Abild-Pedersen, K. W. Jacobsen, T. Bligaard, and J. K. Nørskov, Science **345**, 197 (2014).
[11] R. Christensen, J. S. Hummelshøj, H. A. Hansen, and T. Vegge, J. Phys. Chem. C **119**, 17596 (2015).
[12] M. Pandey and K. W. Jacobsen, Phys. Rev. B **91**, 235201 (2015).
[13] F. Göltl and P. Sautet, J. Chem. Phys. **140**, 154105 (2014).
[14] J. P. Perdew and K. Schmidt, in *Density Functional Theory and its Application to Materials*, Vol. 577, edited by V. Van Doren, C. Van Alsenoy, and P. Geerlings (AIP, New York, 2001), p. 1.
[15] L. Schimka, J. Harl, A. Stroppa, A. Grüneis, M. Marsman, F. Mittendorfer, and G. Kresse, Nat. Mater. **9**, 741 (2010).
[16] S. V. Levchenko, X. Ren, J. Wieferink, R. Johanni, P. Rinke, V. Blum, and M. Scheffler, Comput. Phys. Commun. **192**, 60 (2015).
[17] J. M. del Campo, J. L. Gázquez, S. Trickey, and A. Vela, Chem. Phys. Lett. **543**, 179 (2012).
[18] M. Aldegunde, J. R. Kermode, and N. Zabaras, J. Comput. Phys. **311**, 173 (2016).
[19] L. A. Constantin, E. Fabiano, and F. Della Sala, J. Chem. Theory Comput. **9**, 2256 (2013).
[20] J. Sun, J. P. Perdew, and A. Ruzsinszky, Proc. Natl. Acad. Sci. USA **112**, 685 (2015).
[21] J. Sun, A. Ruzsinszky, and J. P. Perdew, Phys. Rev. Lett. **115**, 036402 (2015).
[22] F. Della Sala, E. Fabiano, and L. A. Constantin, Phys. Rev. B **91**, 035126 (2015).
[23] L. A. Constantin, E. Fabiano, J. M. Pitarke, and F. Della Sala, Phys. Rev. B **93**, 115127 (2016).
[24] H. S. Yu, X. He, and D. G. Truhlar, J. Chem. Theory Comput. **12**, 1280 (2016).
[25] R. Peverati and D. G. Truhlar, J. Phys. Chem. Lett. **3**, 117 (2011).
[26] F. Tran, J. Stelzl, and P. Blaha, J. Chem Phys. **144**, 204120 (2016).
[27] K. Lee, E. D. Murray, L. Kong, B. I. Lundqvist, and D. C. Langreth, Phys. Rev. B **82**, 081101 (2010).
[28] M. Dion, H. Rydberg, E. Schröder, D. C. Langreth, and B. I. Lundqvist, Phys. Rev. Lett. **92**, 246401 (2004).
[29] J. Sun, B. Xiao, Y. Fang, R. Haunschild, P. Hao, A. Ruzsinszky, G. I. Csonka, G. E. Scuseria, and J. P. Perdew, Phys. Rev. Lett. **111**, 106401 (2013).
[30] J. Klimes and A. Michaelides, J. Chem. Phys. **137**, 120901 (2012).
[31] J. Tao, J. P. Perdew, V. N. Staroverov, and G. E. Scuseria, Phys. Rev. Lett. **91**, 146401 (2003).
[32] The t_s transformation is equal to the Padè approximant used in the exchange enhancement function of PBEsol [36].
[33] J. Sun, B. Xiao, and A. Ruzsinszky, J. Chem. Phys. **137**, 051101 (2012).
[34] J. P. Perdew and Y. Wang, Phys. Rev. B **45**, 13244 (1992).
[35] J. P. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996).
[36] J. P. Perdew, A. Ruzsinszky, G. I. Csonka, O. A. Vydrov, G. E. Scuseria, L. A. Constantin, X. Zhou, and K. Burke, Phys. Rev. Lett. **100**, 136406 (2008).
[37] J. P. Perdew, A. Ruzsinszky, G. I. Csonka, L. A. Constantin, and J. Sun, Phys. Rev. Lett. **103**, 026403 (2009).
[38] In addition to better overall prediction power, this should allow for better error estimation when van der Waals forces are important.
[39] Compiled in Ref. [9] based on Refs. [99,100].
[40] L. Grafova, M. Pitonak, J. Rezac, and P. Hobza, J. Chem. Theory Comput. **6**, 2365 (2010).
[41] J. J. Mortensen, L. B. Hansen, and K. W. Jacobsen, Phys. Rev. B **71**, 035109 (2005).
[42] J. Enkovaara, C. Rostgaard, J. J. Mortensen, J. Chen, M. Dułak, L. Ferrighi, J. Gavnholt, C. Glinsvad, V. Haikola, H. A. Hansen, H. H. Kristoffersen, M. Kuisma, A. H. Larsen, L. Lehtovaara, M. Ljungberg, O. Lopez-Acevedo, P. G. Moses, J. Ojanen, T. Olsen, V. Petzold, N. A. Romero, J. Stausholm-Møller, M. Strange, G. A. Tritsaris, M. Vanin, M. Walter, B. Hammer, H. Häkkinen, G. K. H. Madsen, R. M. Nieminen, J. K. Nørskov,

M. Puska, T. T. Rantala, J. Schiøtz, K. S. Thygesen, and K. W. Jacobsen, J. Phys.: Condens. Matter **22**, 253202 (2010).

[43] A. E. Hoerl and R. W. Kennard, Technometrics **12**, 55 (1970).

[44] C. M. Bishop, *Pattern Recognition and Machine Learning*, 1st ed. (Springer, New York, 2006).

[45] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. (Springer, New York, 2009).

[46] The form was chosen as it would make the below linearization convenient. Alternatively, one could use the form $\Phi(\mathbf{a},\omega,\mathcal{W}) = \prod_i L_i(\mathbf{a})^{\mathcal{W}_\mathrm{i}} \cdot R(\mathbf{a},\omega)$, where we would then also transform the regularization in the linearization, hence $\tilde{K}(\boldsymbol{a};\omega) = \sum_i \mathcal{W}_i \frac{L_i(\boldsymbol{a})}{L_i(\boldsymbol{a}_0)} + \frac{R(\boldsymbol{a};\omega)}{R(\boldsymbol{a}_0;\omega)} = \tilde{L}(\mathbf{a},\mathbf{a}_0) + \tilde{R}(\boldsymbol{a};\omega)$, and thereby also introducing a scaling on the regularization term.

[47] B. Efron, J. Am. Stat. Assoc. **78**, 316 (1983).

[48] P. J. Rousseeuw, J. Am. Stat. Assoc. **79**, 871 (1984).

[49] R. Maronna, D. Martin, and V. Yohai, *Robust Statistics: Theory and Methods*, Wiley Series in Probability and Statistics (Wiley, New York, 2006).

[50] The breakdown point is more precisely defined as the maximum proportion of the training data that can be arbitrarily altered while the fitting procedure still chooses a good model for the rest of the training data.

[51] V. J. Yohai, Ann. Stat. **15**, 642 (1987).

[52] P. Rousseeuw and V. Yohai, in *Robust and Nonlinear Time Series Analysis* (Springer, New York, 1984), pp. 256–272.

[53] P. Huber, Ann. Math. Statist. **35**, 73 (1964).

[54] R. A. Maronna, Technometrics **53**, 44 (2011).

[55] For proof of convergence of the ridge-regression S-estimator, see Ref. [54].

[56] We solve with Huber's second method of Ref. [53] as implemented in Statsmodels [101].

[57] A better approach would probably be to use the hierarchical bootstrap samples and find the optimal fits to these with the product loss cost function.

[58] The equation for c_0 is adapted from the source code file PeYoRid accompanying Ref. [54].

[59] This was chosen following the last paragraph of 2.5 in Ref. [54], based on $c_1 = 3.44$, and should have been adjusted to the c_0 value for the SE loss function. However, this is the only influence the regularization strength used to find the initial scale, so we expect this discrepancy to have caused minimal harm.

[60] To get down to this convergence, we mixed the new weights with the weights of the last iteration by $\mathbf{W} = (1 - \mathrm{mix}) \cdot \mathbf{W}_l + \mathrm{mix} \cdot \mathbf{W}_{l-1}$, with $\mathrm{mix} = \min(0.98, 0.98 \cdot \frac{l}{40})$, where l is the iteration number.

[61] This corrects for an underestimation of the true error scale according to Ref. [54]. Compared to its corrections, however, we use $\mathrm{Tr}(\mathbf{W}_i)$ instead of N_i as this is the number of system that we effectively are fitting.

[62] The total clock time of the fitting procedure for the mBEEF-vdW fit was 350 minutes on a single-core Intel Xeon processor from 2013. The clock time can be dramatically reduced with optimization and parallelization.

[63] This also allows us to calculate the singular value decompositions of the design matrices once for all the bootstrap samples and reuse them for all regularization strengths.

[64] For RC1, we used a fitting procedure where the outliers are identified for each dataset independently, and the fit is based on 64 exchange basis functions.

[65] L. A. Constantin, A. Terentjevs, F. Della Sala, P. Cortona, and E. Fabiano, Phys. Rev. B **93**, 045126 (2016).

[66] In the Supplemental Material [68], we provide tables of EPE and err as a function of $M_s, M_\alpha \in [3,10]$, where it can be observed that the lowest EPE and err are found in the diagonal where $M_s = M_\alpha$.

[67] We also provide visualizations of the residuals and $\mathbf{w}$ in the Supplemental Material [68], and a list of the largest outliers for each dataset.

[68] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevB.93.235162 for (1) table comparing the least-squares loss function with the MM-estimator loss function for varying model complexities, (2) figure showing the residuals and robust weights for the training datasets with an accompanying list of the systems that are identified as outliers in each dataset, (3) CO puzzle benchmark, with site preference of molecular CO adsorption on close-packed surfaces of late transition metals for GGA, MGGA, and BEEF family functionals, (4) figure with benchmark on training datasets in mean absolute error statistics, (5) additional bivariate plot figures, (6) the fitting coefficients of mBEEF–vdW and the ensemble matrix in comma-separated values (CSV) files, (7) S22x5 binding curve figures, (8) Bayesian error estimation statistics for training datasets (9) CSV files containing statistics used for benchmark figures, and (10) CSV files containing the raw DFT data used for benchmarking.

[69] Additional information can be found in the Supplemental Material [68], including the mean absolute error of RMSE and BEE estimates, and comparison of self-consistent to non-self-consistent results.

[70] Using the values of Table IV of Ref. [9], we find the geometric mean of RMSE prediction errors for BEEF-vdW to be 29 meV, whereas the geometric mean of the BEE ensemble estimates RMSE is 90 meV.

[71] B. Hammer, L. B. Hansen, and J. K. Nørskov, Phys. Rev. B **59**, 7413 (1999).

[72] L. Goerigk and S. Grimme, J. Chem. Theory Comput. **6**, 107 (2010).

[73] J. Klimes, D. R. Bowler, and A. Michaelides, J. Phys.: Condens. Matter **22**, 022201 (2010).

[74] V. R. Cooper, Phys. Rev. B **81**, 161104(R) (2010).

[75] F. Mittendorfer, A. Garhofer, J. Redinger, J. Klimes, J. Harl, and G. Kresse, Phys. Rev. B **84**, 201401(R) (2011).

[76] G. Bertoni, L. Calmels, A. Altibelli, and V. Serin, Phys. Rev. B **71**, 075402 (2005).

[77] G. Kalibaeva, R. Vuilleumier, S. Meloni, A. Alavi, G. Ciccotti, and R. Rosei, J. Phys. Chem. B **110**, 3638 (2006).

[78] G. Giovannetti, P. A. Khomyakov, G. Brocks, V. M. Karpan, J. van den Brink, and P. J. Kelly, Phys. Rev. Lett. **101**, 026803 (2008).

[79] M. Fuentes-Cabrera, M. I. Baskes, A. V. Melechko, and M. L. Simpson, Phys. Rev. B **77**, 035405 (2008).

[80] M. Vanin, J. J. Mortensen, A. K. Kelkkanen, J. M. Garcia-Lastra, K. S. Thygesen, and K. W. Jacobsen, Phys. Rev. B **81**, 081408 (2010).

[81] I. Hamada and M. Otani, Phys. Rev. B **82**, 153412 (2010).

[82] C. Gong, G. Lee, B. Shan, E. M. Vogel, R. M. Wallace, and K. Cho, J. Appl. Phys. **108**, 123711 (2010).

[83] T. Olsen and K. S. Thygesen, Phys. Rev. B **87**, 075111 (2013).

[84] M. Andersen, L. Hornekær, and B. Hammer, Phys. Rev. B **86**, 085405 (2012).

[85] Y. Gamo, A. Nagashima, M. Wakabayashi, M. Teria, and C. Oshima, Surf. Sci. **374**, 61 (1997).

[86] G. Román-Pérez and J. M. Soler, Phys. Rev. Lett. **103**, 096102 (2009).

[87] V. U. Nazarov and G. Vignale, Phys. Rev. Lett. **107**, 216402 (2011).

[88] M. A. Marques, M. J. Oliveira, and T. Burnus, Comput. Phys. Commun. **183**, 2272 (2012).

[89] G. Kresse and J. Hafner, Phys. Rev. B **47**, 558 (1993).

[90] G. Kresse and J. Hafner, Phys. Rev. B **49**, 14251 (1994).

[91] G. Kresse and J. Furthmüller, Comput. Mater. Sci. **6**, 15 (1996).

[92] G. Kresse and J. Furthmüller, Phys. Rev. B **54**, 11169 (1996).

[93] P. Giannozzi, S. Baroni, N. Bonini, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, G. L. Chiarotti, M. Cococcioni, I. Dabo, A. Dal Corso, S. de Gironcoli, S. Fabris, G. Fratesi, R. Gebauer, U. Gerstmann, C. Gougoussis, A. Kokalj, M. Lazzeri, L. Martin-Samos, N. Marzari, F. Mauri, R. Mazzarello, S. Paolini, A. Pasquarello, L. Paulatto, C. Sbraccia, S. Scandolo, G. Sclauzero, A. P. Seitsonen, A. Smogunov, P. Umari, and R. M. Wentzcovitch, J. Phys.: Condens. Matter **21**, 395502 (2009).

[94] J. Rezac, K. E. Riley, and P. Hobza, J. Chem. Theory Comput. **7**, 2427 (2011).

[95] P. Janthon, S. Luo, S. M. Kozlov, F. Viñes, J. Limtrakul, D. G. Truhlar, and F. Illas, J. Chem. Theory Comput. **10**, 3832 (2014).

[96] J. Wellendorff, T. L. Silbaugh, D. Garcia-Pintos, J. K. Nørskov, T. Bligaard, F. Studt, and C. T. Campbell, Surf. Sci. **640**, 36 (2015).

[97] J. P. Perdew and A. Zunger, Phys. Rev. B **23**, 5048 (1981).

[98] V. I. Anisimov, J. Zaanen, and O. K. Andersen, Phys. Rev. B **44**, 943 (1991).

[99] J. A. Pople, M. Head-Gordon, D. J. Fox, K. Raghavachari, and L. A. Curtiss, J. Chem. Phys. **90**, 5622 (1989).

[100] L. A. Curtiss, K. Raghavachari, G. W. Trucks, P. C. Redfern, and J. A. Pople, J. Chem. Phys. **94**, 7221 (1991).

[101] S. Seabold and J. Perktold, in *Proceedings of the 9th Python in Science Conference*, edited by Stéfan van der Walt and Jarrod Millman (SciPy, Austin, Texas, 2010), pp. 57–61.