

Extrapolation to the complete basis-set limit in density-functional theory using statistical learning

Daniel T. Speckhard^{1,2,*} Christian Carbogno² Luca M. Ghiringhelli^{1,2,†} Sven Lubeck¹
Matthias Scheffler² and Claudia Draxl¹

¹Physics Department and IRIS Adlershof, Humboldt-Universität zu Berlin, Zum Großen Windkanal 2, Berlin 12489, Germany

²NOMAD Laboratory at the Fritz Haber Institute of the Max Planck Society and IRIS Adlershof, Faradayweg 4-6, Berlin 14195, Germany



(Received 15 April 2024; accepted 17 December 2024; published 13 January 2025)

The numerical precision of density-functional-theory (DFT) calculations depends on a variety of computational parameters, one of the most critical being the basis-set size. The ultimate precision is reached in the limit of a complete basis set (CBS). Our aim in this work is to find a machine-learning model that extrapolates finite basis-size calculations to the CBS limit for periodic crystal structures. We start with a data set of 63 binary solids investigated with two all-electron DFT codes, `exciting` and `FHI-aims`, which employ very different types of basis sets. A quantile-random-forest model and a symbolic regression approach using the SISO model are used to estimate the total-energy correction with respect to a fully converged calculation as a function of the basis-set size. The random-forest model achieves a symmetric mean absolute percentage error of lower than 25% for both codes and outperforms previous approaches in the literature. SISO outperforms the random forest model for the `exciting` code. Our approach also provides prediction intervals, which quantify the uncertainty of the models' predictions.

DOI: [10.1103/PhysRevMaterials.9.013801](https://doi.org/10.1103/PhysRevMaterials.9.013801)

I. INTRODUCTION

The assessment of the quality of density-functional-theory (DFT) calculations concerns the accuracy of the exchange-correlation functional and the numerical precision that depends on a variety of computational parameters. To obtain a DFT result as precise as possible for the chosen exchange-correlation (xc) functional, one needs to reach the complete basis-set (CBS) limit [1]. In practice, the basis set is truncated at a size that balances precision and computational cost. Extrapolation from low-precision settings to the CBS limit is commonplace in quantum chemistry [2]. A similar approach has also been applied to DFT calculations for isolated systems [3,4]. In materials science, convergence tests with respect to the basis-set size are typically done, but extrapolation to the CBS limit is not often performed. In contrast to pure extrapolation of computed values, our aim is to statistically learn a model that can extrapolate DFT results to the CBS limit. More specifically, we seek to predict the difference in the total energy per atom computed with an incomplete basis-set size to a computation performed in the CBS limit. We exemplify our approach with a dataset of binary materials.

Note, the extrapolation to the CBS limit depends on the chosen functional. Here, we only consider the PBE xc functional of the generalized gradient approximation (GGA) [5].

Our motivation is twofold. First, such a model opens up the possibility to perform less precise, and therefore computationally less demanding *ab initio* calculations to predict a more precise result. Here, we recall that typical DFT-GGA implementations scale with order $O(N^3)$ where N is the basis-set size. Second, it allows us to assign uncertainty estimates to the huge amounts of *ab initio* data contained in open-access databases. For instance, the Novel Materials Discovery (NOMAD) infrastructure [6] currently hosts over 140 million ground-state DFT calculations. These calculations were carried out for a variety of purposes ranging from molecular-dynamics simulations of complex systems with less precise settings to ultra-high-precision calculations for elemental solids. Uncertainty estimates for the total energies would provide users useful information about the precision of these calculations and how the data can be reused/repurposed [7]. If one can do CBS extrapolation, one could even extrapolate all these data to the CBS limit, which would be even more useful.

In this work, we train statistical learning models to predict the total-energy difference ΔE^{AB} for binary materials containing the two elements A and B . We train on a dataset consisting of DFT results for 71 elemental and 63 binary solids, computed by the two full-potential all-electron codes `FHI-aims` and `exciting` with varying basis-set sizes. For details concerning the dataset we refer to Ref. [8]. The linearized augmented planewave (LAPW) code `exciting` employs augmented planewaves (APW) plus local orbitals (LO) as its basis, while `FHI-aims` uses numeric atom-centered orbitals (NAOs). These two codes are representatives of all-electron,

*Contact author: daniel.speckhard@physik.hu-berlin.de

†Present address: Department of Materials Science and Engineering, Friedrich-Alexander Universität Erlangen-Nürnberg, Germany.

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI. Open access publication funded by Max Planck Society.

full potential packages. This means that they can simulate the behavior of all electrons in a material on the same footing and are proven to be among the most precise DFT codes available [9]. In the Hartree-Fock framework, Gaussian-type orbitals (GTOs) follow an exponential decay function with respect to the square root of the number of GTOs used [10]. While for NAOs used in FHI-aims, also a roughly exponential decay behavior was observed in Ref. [8], for the LAPW basis set as implemented in *exciting*, the logarithm of the total energy showed a roughly linear relationship with number of plane waves used. Despite the significantly different concepts, algorithms, and numerical approaches, both codes are expected to give close-to identical results [9,11]. Due to the very different basis sets and thus numerical implementations, we investigate the behavior of their convergence separately. We compare our modeling efforts using a quantile-random forest (QRF) and the SISO algorithm to a stoichiometric model which was introduced in Ref. [8] and find that our models outperform the latter in terms of the relevant metrics. We provide an accompanying Jupyter notebook for the user to explore the training data and to train or repurpose the models presented in this work.

II. METHODS

The task of extrapolation to the CBS limit is formulated as a Δ -learning problem [12]. For results of a single DFT calculation at a fixed basis-set size, N_b , which is smaller than that of the converged case, the CBS limit is denoted as N_∞ . The calculations underlying the dataset in Ref. [8], were performed by increasing the basis-set size until the total energy converged within 10^{-4} eV/atom, which we define as the CBS limit. Note, however, that within the self-consistent DFT calculations, the total-energy convergence criteria was much smaller than the CBS limit so as not to affect the total-energy dependence on the basis-set size. The data is fed into a statistical learning algorithm to estimate the difference between the imprecise DFT calculation and the CBS limit. Our task, in other words, is to estimate the total energy per atom of a binary material composed of elements A and B , in the CBS limit, $E^{AB}(N_\infty)$, using the results of a DFT calculation with the fixed incomplete basis-set size, N_b . Mathematically, we aim at finding the change (Δ) in total-energy from an incomplete basis-set size, $E^{AB}(N_b)$, to the CBS limit $E^{AB}(N_\infty)$. As can be seen in Eq. (1), we target $\Delta E^{AB}(N_b)$.

$$E^{AB}(N_\infty) = E^{AB}(N_b) + \Delta E^{AB}(N_b). \quad (1)$$

We employ different models for obtaining the total-energy differences ΔE^{AB} per atom. Other DFT settings are kept constant. Our aim is to use a less computationally intense and thus less precise calculation to obtain $E^{AB}(N_b)$ in tandem with a statistically learned model that predicts $\Delta E^{AB}(N_b)$. Eventually, we predict the total energy of the complete basis-set limit.

Our baseline model, to compare our new approach with, is a stoichiometric model introduced in Ref. [8],

$$\Delta E^{AB}(N_b) = C^A * \Delta E^A(N_b) + C^B * \Delta E^B(N_b). \quad (2)$$

Each binary solid, represented as AB , is composed of two chemical elements, labeled A and B . Here, the letter A (B)

refers to the less (more) electronegative element in the binary. $\Delta E^A(N_b)$ refers to the CBS total-energy correction for the elemental solid, containing only element A , when using a basis-set size N_b . The C^A is the stoichiometric fraction that the element A appears in the binary. The CBS corrections, $\Delta E^A(N_b)$, in Ref. [8] were calculated for the elemental solids in the same manner as the binaries, i.e., the basis-set size was increased until the total-energy CBS limit was reached.

A. Model features and basis sets

We feed the QRF and SISO models information about the two elements in terms of elemental solids by providing $C^A * \Delta E^A(N_b)$ and $C^B * \Delta E^B(N_b)$, which are the two terms in the stoichiometric model from Eq. (2). Atomic information about the elements in terms of isolated atoms is also provided. Their use is motivated by other statistical-learning models in materials science [13]. The electron affinity (EA^A , EA^B), the ionization energy (EI^A , EI^B) and the mean radius (r_s^A , r_s^B) for the s orbital of element A/B computed with FHI-aims are also fed into the two models.

The most important parameter determining the quality of augmented plane-wave basis sets is $RK_{\max}$, which is the product of the radius of the smallest atomic (muffin-tin) sphere and the plane-wave cutoff. In Ref. [8], a precision factor, $(RK_{\max}/RK_{\max}^{\text{opt}})^2$ was introduced, which denotes the precision of the basis set of *exciting*. The dataset we use contains elemental solids and binaries at the same percentage value of the precision factor. However, $RK_{\max}^{\text{opt}}$ may be different for a binary and its elemental solids. Note, in this dataset, the number of APWs is varied but the number of LOs is kept constant. More information about the basis-set precision parameter for *exciting* is given in the supplementary information of Ref. [8].

FHI-aims offers tabulated species-specific suggestions for numerical settings and NAOs, named as “light,” “tight,” or “really tight” defaults. In general, one does not need to use the tabulated settings. However, in this work we do. On top of the numerical settings defined in these defaults, we also consider different basis-set size settings (minimal, standard, tier1, or tier2). The combination of both ultimately dictates the number and type (s , p , d , etc.) of basis functions included in a calculation. “Standard” refers to the default basis-set size suggested in the respective numerical setting. The difference in the number of NAOs per valence electron from the CBS limit, labeled ΔSB_{PVE}^{AB} , is used as a basis-set size metric. More information about the basis-set size in FHI-aims is given in the supplementary information of Ref. [8].

B. Metrics

Our DFT data used for the training contain CBS energy corrections, ranging in magnitude from 10^{-6} eV/atom to several eV/atom. The root-mean-squared-error (RMSE) and mean-absolute-error (MAE), which are common loss functions, are known to give more weight to larger targets in datasets that span several orders of magnitude [14]. Therefore, we optimize our QRF models for the root-mean-squared-logarithmic-error (RMSLE) to best capture logarithmic-scale differences in the target. The RMSLE as defined in Ref. [15],

is given by

$$\text{RMSLE} = \sqrt{\frac{1}{N} \sum_{i=1}^N \{\log[y_i(\vec{x}_i) + \epsilon] - \log[h(\vec{x}_i) + \epsilon]\}^2}, \quad (3)$$

where the $\vec{x}_i$ are the feature vectors (combinations of feature values), the $y_i(\vec{x}_i)$ are the calculated CBS energy corrections to be predicted, and $h(\vec{x}_i)$ are our predictions from the statistical learning model h . As the target should be positive valued, and the values of $\Delta E^{AB}(N_b)$ are always negative due to the variational principle, we set $y_i(\vec{x}_i) = -\Delta E^{AB}(N_b)$. The addition of a small value ϵ in the logarithm argument serves to avoid a zero as the logarithmic argument. With a Taylor expansion for $x \ll 1$, a choice of $\epsilon = 1$ yields $\log(x + 1) \approx x$, and we arrive back at a metric similar to the MAE that gives more weight to larger targets. Since our CBS energy convergence criteria is 10^{-4} eV/atom, we choose this value as ϵ .

Another popular metric for targets that vary orders of magnitude is the symmetric mean absolute percentage error (sMAPE) [14]:

$$\text{sMAPE} = \sum_{i=1}^N \frac{|h(\vec{x}_i) - y_i(\vec{x}_i)|}{\frac{1}{2}|y_i(\vec{x}_i)| + \frac{1}{2}|h(\vec{x}_i)|} \times 100. \quad (4)$$

It is closely related to the mean absolute percentage error (MAPE), defined with a different denominator containing the target as:

$$\text{MAPE} = \sum_{i=1}^N \frac{|h(\vec{x}_i) - y_i(\vec{x}_i)|}{|h(\vec{x}_i)|} \times 100. \quad (5)$$

The sMAPE is often employed to optimize machine-learning models operating on time series data where the target can grow exponentially [16]. While the MAPE is unbounded from above, the sMAPE is at most 200% when either the target or prediction is zero. This means that the sMAPE avoids issues when the target $y_i(\vec{x}_i)$ is close to zero where the MAPE tends to diverge [14]. We also consider the MAPE as a metric, since it is easier to comprehend. For completeness, we include the MAE and the 95% quantile absolute error (95% quantile metric for short) as metrics. The latter represents the 95% quantile of the absolute error (AE) for each model.

C. Quantile random forests

We choose to use quantile random forests (QRF) for CBS extrapolation since they combine the benefits of random forests (RF) and provide the user with uncertainty estimates [17]. RFs are known to perform well on a wide range of tasks and require minimal tuning and no data scaling [18]. Here, we provide a brief summary of decision trees, the random forest algorithm, and the quantile random-forest method introduced by Meinshausen [17]. Random forests are a collection of decision tree models. A decision tree is a piecewise constant model, where the tree partitions the input feature space into discrete regions. The tree then assigns a constant estimate for each region depending on the training data points that fall in that region [19]. The decision-tree model can be described by

the following equation:

$$f(\vec{x}_i) = \sum_{m=1}^M c_m I(\vec{x}_i), \quad (6)$$

where the indicator function $I(\vec{x}_i)$ is one when $\vec{x}_i \in R_m$ and zero otherwise, and c_m is the corresponding constant prediction for region R_m of the input feature space. The most common algorithm subdivides with each split the latter into two nonoverlapping regions. It then continues to sequentially partition the input feature space (one partition at a time) until a stopping condition is met (e.g., minimum number of training datapoints in each region or further splits do not reduce the loss metric). Once the decision tree is built, inference occurs by examining the region in which the input data point falls and then retrieving the constant for that region to be used as the prediction.

The random-forest algorithm suffers less from overfitting than a single decision tree [20]. A random forest consist of a collection of diverse decision-tree models. The diversity is created through two strategies. First, each decision tree is trained on a slightly different training dataset, which are created by sampling the training data with replacement (bootstrapping) [21]. Second, the algorithm forces each split for each decision tree to be done on a random subset of model features. To make an inference, the random forest uses the leaf node for each tree, i.e., the region into which the input data point falls, and takes the average prediction from the collection of trees. The QRF creates quantiles (e.g., 2.5% and 97.5%) by sorting the training data that fall in the leaf nodes. When a prediction is to be made, at every leaf node, the training-data targets are gathered (repeated values are included), sorted from smallest to largest, and the prediction quantiles are created. These quantiles are used as statistically meaningful prediction intervals. These prediction intervals provide an estimate of the uncertainty of the model upon making an inference. In this work, we apply the QRF method in a regression setting to minimize the RMSLE metric. In the Appendix we show how decision-tree parameters are learned when optimizing for this metric.

D. SISSO model

The sure independence screening and sparsifying operator (SISSO) method [22] is briefly outlined here, while more details can be found in Ref. [23]. This model combines features and a set of operators (e.g., exponentiation, division, multiplication, square root) to build new features, called secondary features. The number of secondary features exhibits combinatorial growth with increasing the number of input features and the set of mathematical operations. To control the final feature space, secondary, highly nonlinear features are screened by looking at the correlation between them and the model target. Only the top performing secondary features (in our case the top 100 secondary features) are then included in the next step. The maximum dimension of secondary features (provided as input) are combined linearly. The SISSO model fits a linear model using an ℓ_0 penalty to strictly enforce the number of secondary features in the resulting model. The ℓ_0 regularization is combinatorial in complexity.

TABLE I. Metrics for the QRF and the SISSO models for **exciting** and FHI-aims on held-out test data. The stoichiometric model performance is shown for comparison. The 95% quantile metric refers to the 95% quantile of the absolute error between the model and the DFT CBS corrections. The corresponding training data metrics are shown in parentheses. Note that **exciting** data contains larger calculated DFT ΔE^{AB} targets since the basis-set size was controlled manually whereas the FHI-aims basis variation stopped at the discrete *minimal* option given by the code. This results in larger MAE and maximum error values for the **exciting** models.

Metric	exciting			FHI-aims		
	Stoichiometric	QRF	SISSO	Stoichiometric	QRF	SISSO
sMAPE (%)	28.7 (29.6)	24.2 (10.6)	23.6 (19.9)	48.9 (53.3)	13.9 (10.6)	39.9 (45.7)
MAPE (%)	27.9 (30.6)	27.3 (11.6)	23.6 (20.8)	58.2 (70.1)	14.9 (29.0)	45.4 (75.8)
RMSLE	0.383 (0.376)	0.318 (0.150)	0.307 (0.261)	0.663 (0.916)	0.218 (0.213)	0.580 (0.621)
MAE (eV/atom)	1.556 (1.250)	1.437 (0.313)	1.296 (0.590)	0.257 (0.212)	0.036 (0.013)	0.207 (0.178)
95% quantile (eV/atom)	5.900 (8.018)	7.795 (1.584)	6.004 (3.272)	1.520 (1.420)	0.256 (0.072)	0.141 (0.117)

We use the SISSO++ library and train on GPUs to expedite training [24,25]. We observe training times around twelve hours on four Nvidia V100 GPUs. As available SISSO software packages do not have the RMSLE metric implemented for optimization, we optimize the SISSO model using the RMSE, with a modified target, which is summarized in the optimization equation:

$$\min_{h \in H} \sqrt{\frac{1}{N} \sum_{i=1}^N \{\log[\Delta E(\vec{x}_i) + \epsilon] - h(\vec{x}_i)\}^2}. \quad (7)$$

The best performing SISSO model, $h(\vec{x}_i)$ from the possible model space H , is the one that minimizes the RMSE of the new target $\log[\Delta E^{AB}(\vec{x}_i) + \epsilon]$. With $h(\vec{x}_i) = \log[g(\vec{x}_i) + \epsilon]$, the above optimized metric is the same as the RMSLE for the model $g(\vec{x}_i)$. We use the latter for inference (and metrics) on the original target, $\Delta E^{AB}(N_b)$.

E. Training and test splits

Our data are randomly split into training and test data using an 80/20 split. We perform tenfold grid-search cross validation (CV) to choose the number of decision trees in the forest, the minimum number of samples per leaf, and the fraction of features considered at each split. More details on the CV can be found in the Appendix and the accompanying Jupyter notebook [26].

III. RESULTS

We analyze the performance of three models on the data, namely the stoichiometric model, the QRF model and the SISSO model. Table I summarizes the most relevant metrics obtained for the test and training data. Note that the models are trained separately on FHI-aims and **exciting** data. Recall also that the stoichiometric model has no free parameters to learn via training on the data.

The QRF models perform better than the stoichiometric models for all metrics except the 95% quantile of absolute errors where it does slightly worse for **exciting**. In general, the CBS energy corrections contain larger values for **exciting** compared to FHI-aims, since the LAPW nature allows for manually reducing the basis-set size to close to zero [8]. We see this fact in the standard deviation of the CBS energy corrections of the test data, which is 8.127 and 0.738 eV/atom

for **exciting** and FHI-aims, respectively. The larger 95% quantile of absolute errors indicates that the stoichiometric model does slightly better than the QRF when targeting very large CBS energy corrections on the order of several eV/atom. For the MAPE metric in the FHI-aims case, surprisingly, the QRF model performs slightly better in the test data than in the training data. This may be due to the fact the QRF model is trained to fit the RMSLE metric and not the MAPE metric. In contrast, we do not see a larger training error for the other five metrics summarized in Table I. The SISSO model for **exciting** shows slight improvement with respect to the QRF model in terms of all metrics. The FHI-aims SISSO model falls in between the stoichiometric and QRF model.

The QRF models predictions are plotted against the DFT targets in Fig. 1. Note that logarithmic scales are used to visually capture several orders of magnitude in the total-energy corrections. Relevant metrics, the sMAPE and MAE, are shown to help the reader understand the quality of fit. The RMSLE is also shown as the metric that is optimized during training.

We observe that the stoichiometric model for FHI-aims does quite poorly (test sMAPE of 48.9%) whereas the stoichiometric model performs better on the **exciting** dataset with an sMAPE of 28.7%. This is understood as the LAPW basis of **exciting** is gradually improved by increasing the basis-set size parameter $RK_{\max}$. The atomic-centered orbitals used in FHI-aims, on the other hand, give rise to a discrete basis-set size, specified as tiers (minimal, tier 1, tier 2), which correspond to more abrupt changes in basis-set quality. However, our nonlinear QRF model provides a pretty good model for the CBS limit of FHI-aims. The discrete and piecewise nature of its basis sets may explain why the very nonlinear and piecewise nature of the RF models succeeds on the FHI-aims dataset.

The SISSO models provide the user with closed-form equations that can be interpreted. The best performing SISSO model for the **exciting** data is

$$\begin{aligned} & \log[\Delta E^{AB}(N_b) + \epsilon] \\ &= a_0 \log[C^B \Delta E^B(N_b) + \alpha_0 * C^A \Delta E^A(N_b) + \beta_0] \\ &+ a_1 \frac{e^{-\alpha_1 * \text{EI}^B}}{\text{EA}^B + \text{EI}^A} + a_2 \frac{e^{-\alpha_2 * \text{EI}^B}}{(\text{EA}^A + \beta_2)^3} + a_3. \end{aligned} \quad (8)$$

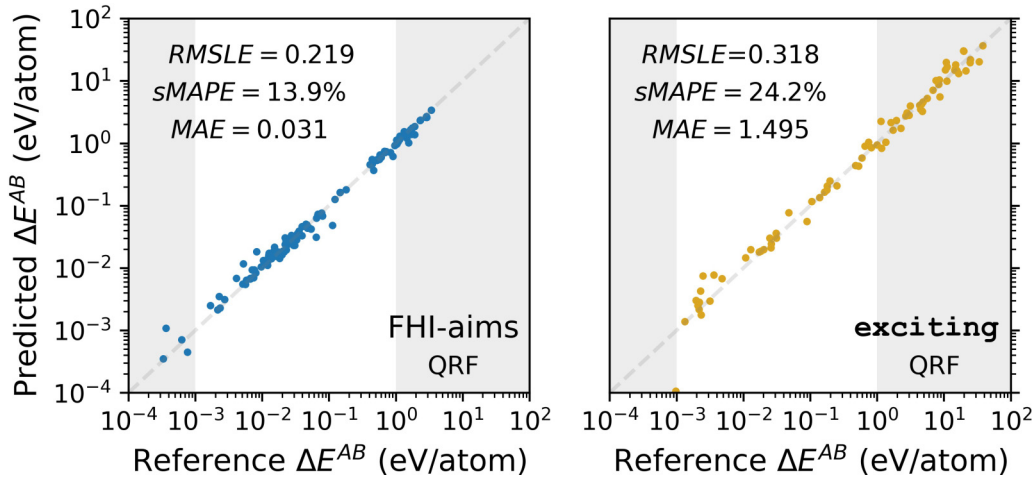


FIG. 1. Predictions of total-energy differences of the QRF model for FHI-aims (left) and *exciting* (right) data, plotted against the respective DFT results. Relevant error metrics are added to help interpret the quality of the fit. Note the logarithmic axes. The region of DFT calculated ΔE^{AB} values between 1 meV/atom and 1 eV/atom is plotted with a lighter shade since these data are of particular interest to DFT practitioners.

We immediately see that the term $C^B \Delta E^B(N_b) + \alpha_0 * C^A \Delta E^A(N_b) + \beta_0$ resembles the stoichiometric model. With α_0 being 0.47, of $C^B \Delta E^B(N_b)$ of the more electronegative

elemental solid is given more weight in comparison to $C^A \Delta E^A(N_b)$. The best performing SISSO model for the FHI-aims data is

$$\log[\Delta E^{AB}(N_b) + \epsilon] = a_0 \log[C^B \Delta E^B(N_b) + \alpha_0 * C^A \Delta E^A(N_b)] + a_1 \frac{EA^B EA^A}{(C^A \Delta E^A + \alpha_1)^{1/3}} + a_2 \frac{EA^B}{r_s^B} (C^A \Delta E^A + \alpha_2)^{-1/2} + a_3. \quad (9)$$

Here, the term $E^B(N_b) + \alpha_0 * \Delta E^A(N_b)$ is similar to both the *exciting* SISSO model and the stoichiometric model. Like the *exciting* SISSO model, α_0 is less than one (0.78). For both SISSO models, EA^B is present, a feature also used by the FHI-aims QRF model. The model performance mirrors that of the stoichiometric model. Due to the similarity of the first term, this is not surprising. In the Appendix (Fig. 8), we see that the term resembling the stoichiometric model is much more important than the other terms in reducing the RMSLE. In the *exciting* SISSO model, all of the terms help to reduce the RMSLE significantly, although the term resembling the stoichiometric model is the most important. Both models perform well. Therefore, also for *exciting* the SISSO model mirrors the performance of the stoichiometric model.

The violin plots in Fig. 2 display the distributions of symmetric percentage errors (SPE) on the test data for all models. Violin plots [28] combine kernel-density plots with box plots, where the latter shows the model's SPE quantiles (5%, 25%, 50%, 75%, 95%). Outliers are plotted with dots above the respective 95% level. Note, the 95% quantile of SPE is not the same as the 95% quantile of AE metric in Table I. The kernel-density plots, underneath the box plots, provide estimates for the probability-density for the SPE, e.g., they estimate the likelihood of the prediction errors in a given range when using the model. Note, the density plots assume the underlying distribution is Gaussian, which may not be true

but is done to help the user visualize how the distribution might look like. For FHI-aims, the QRF model concentrates the SPE around 10% while the stoichiometric model shows a corresponding thin distribution spanning a much larger range of SPE. The SISSO model shows a median SPE smaller than the QRF model for *exciting* and a larger one for FHI-aims.

The *exciting* QRF model has a smaller median 25% quartile and 95% quartile than the stoichiometric model but the 75% quartile of the QRF model is larger. The larger 95% quartile of the stoichiometric model is likely the reason why the stoichiometric model returns a significantly worse sMAPE. This means that the QRF model has less very large SPE, which is a desired behavior. The SISSO model seems to share the outlier behavior of the stoichiometric model with larger SPE outlier values than the QRF model that appear similar in value to the stoichiometric model. The SISSO model shows more datapoints outside the 95% quantile and in absolute terms a larger maximum SPE than the QRF model. No meaningful trends can be identified for the QRF or SISSO models for the predictions on test data of both codes that returned SPE above the 95% quantile.

The Gini importance, or mean decrease of impurity (MDI) [20], of the features fed into the QRF models for both codes is displayed in Fig. 3. The figure quantifies, for each variable, the QRF's ability to reduce the optimization metric using splits on that feature. Both codes' QRF models strongly depend primarily on either $\Delta E^A(N_b)$ or $\Delta E^B(N_b)$, which are the two

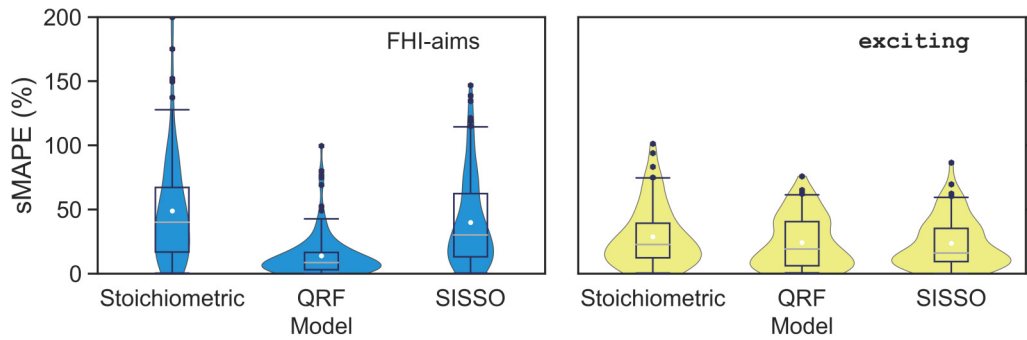


FIG. 2. Distribution of symmetric percentage errors of the stoichiometric, QRF, and SISSO models on held-out test data for FHI-aims (left) and `exciting` (right) as violin plots. The box plot on top shows the 5%, 25%, 50%, 75%, and 95% quantiles. The white dot indicates the mean symmetric percentage error, the black dots values that fall outside of the 5% and 95% quantiles. We do not see the 5% quantiles in the figure or data points that are smaller than the 5% quantile since the 5% quantiles are close to zero.

features of the stoichiometric model. That these variables are effective in describing the CBS energy correction of the binary compounds comes as no surprise considering the overall success of the stoichiometric model [8]. For `exciting` $C^B \Delta E^B(N_b)$ has a larger feature importance than $C^A \Delta E^A(N_b)$, similar to the `exciting` SISSO model where $C^B \Delta E^B(N_b)$ is given more weight. The QRF model for FHI-aims, unlike the SISSO model, gives more importance to $C^A \Delta E^A(N_b)$ and this may partially explain why it performs better. EA^B , the electron affinity of the less electronegative element, B , in the binary, is the second and fourth most important feature for the FHI-aims and `exciting` QRF models.

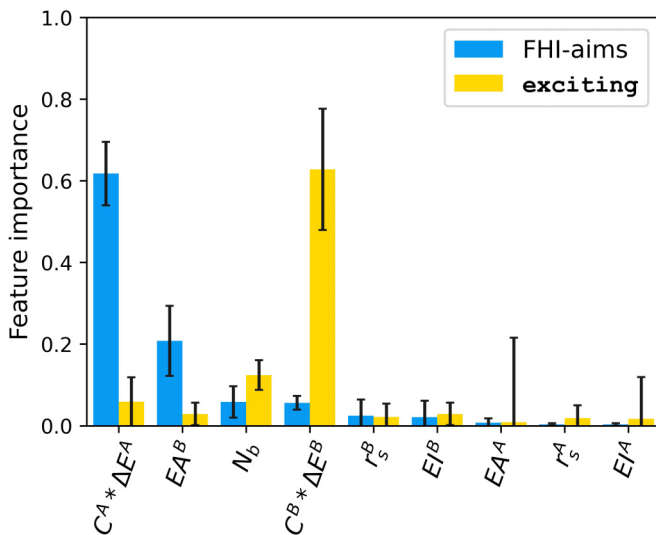


FIG. 3. Feature (Gini) importances of the QRF models for FHI-aims and `exciting`. Black lines show the standard deviation in the feature importance across trees in the forests. The features are ordered in terms of their importance for the FHI-aims QRF model. $C^A * \Delta E^A$ and $C^B * \Delta E^B$ are the first and second terms in the stoichiometric model, respectively. EA^B and EI^B , respectively, are the electron affinity and ionization potential of the elemental solid composed of element B , computed with FHI-aims. r_s^B is the mean radius for the s -like pseudo orbital of elemental solid B (computed with FHI-aims). N_b is the basis-set size for the respective code.

The basis-set size variable, N_b , turns out to be the second and fourth most selected feature for `exciting` and FHI-aims, respectively. For FHI-aims, the basis-set size variable, ΔSB_{PVE}^{AB} , is a single scalar. It maps an s -like orbital and a d -like orbital with the same weight. This loss of information in the mapping may be why we do not see the feature playing an important role. We experimented with feeding the model additional basis-set size variables, namely, the numerical and basis-set size settings encoded as integers but saw no improvement in CV performance. The fact that the features $\Delta E^A(N_b)$ and $\Delta E^B(N_b)$ are functions of the basis-set size may explain why the basis-set size variables themselves are not important since it is implicitly included in these variables.

The improvement of the QRF models over the corresponding stoichiometric models comes from two sources, the ability to use more variables and the ability to express nonlinear functions. Based on the non-negligible feature importance of the basis-set size variable and atomic chemical data such as the electron affinity, we learn that the added features are important. However, this is not the only source of improvement. To provide evidence for it, we trained a linear model (ℓ -1 and ℓ -2 regularization) with these additional variables and found no significant improvement over the stoichiometric model. We do, however, see an improvement with the SISSO model for both codes using nonlinear combinations of primary features. Therefore, we conclude that the QRF model benefits also from its nonlinear nature.

We also trained a QRF model on the residual of the stoichiometric model. It performs well for `exciting` but poor for FHI-aims. The results and analysis of this model can be found in the Appendix. Moreover, we also trained a neural network (NN) due to their use in a wide range of applications [29,30]. Since NNs are data-hungry models, for both datasets the NN performs worse than the QRF and SISSO models, as expected. The NN training procedure (i.e., grid search over number of nodes, layers, batch size, learning rate, dropout rate) and results can be seen in the Appendix and Jupyter notebook.

QRF models not only provide predictions but also associated prediction intervals based on the training data. Prediction intervals provide the user with an estimate for the uncertainty of the QRF prediction [31]. When introducing quantile random forests, Meinshausen [17] uses

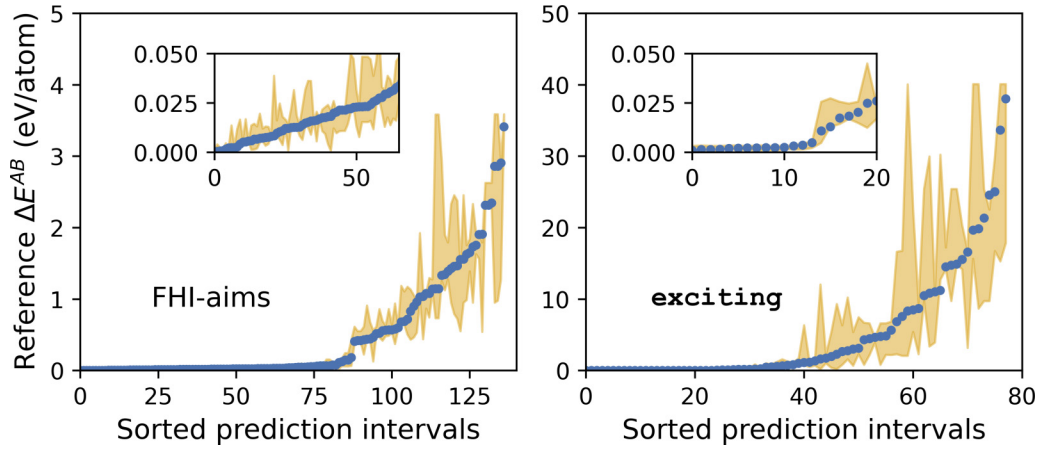


FIG. 4. Prediction intervals (95%) for the QRF models for FHI-aims (left) and `exciting` (right). The blue points indicate the calculated ΔE^{AB} DFT values. The yellow shaded areas show the prediction intervals for each DFT value target (blue point). The data (DFT values and associated prediction intervals) are ordered from left to right in terms of increasing DFT values. The insets zoom into the respective region of smaller values. Note, the x axes are different for both codes since there is varying amount of data for each code due to the different basis-set size parameters. As a result, the y axes are also different. `exciting` contains larger calculated DFT ΔE^{AB} values since the basis-set size was controlled manually whereas the FHI-aims code basis variation stopped at the discrete minimal option given by the code.

95% prediction intervals, which are the difference between the 2.5% quantile and 97.5% quantile. The target CBS energy corrections and their associated 95% prediction intervals are shown in Fig. 4. They contain 92.7% of the held-out test data for FHI-aims and 85.9% for `exciting`. If the estimated 2.5% and 97.5% quantiles from our model were correct, we would expect 95% of the data to fall in this range. In other words, on average 2.5% of all unseen data points from the test dataset should be above and below the 95% prediction interval. This implies that our prediction intervals for both best performing models cover slightly less of the test data than desired. That said, we find the deviation acceptable with only 10 and 11 data points for FHI-aims and `exciting`, respectively, being outliers. We note in passing that 99% prediction intervals (or other values) can also be easily created using the accompanying Jupyter notebook if the user desires more conservative uncertainty estimates.

We further analyze the 95% prediction intervals by plotting them against the calculated DFT values (the model target) in Fig. 5. In general, we would expect larger calculated ΔE^{AB} values to have larger associated model prediction intervals. This is indeed what we see. The `exciting` prediction intervals have a Pearson correlation of 0.73 with the calculated ΔE^{AB} values. For FHI-aims, the correlation of the prediction intervals is lower although still significantly positive at 0.69.

IV. DISCUSSION

The QRF models allow us to perform CBS extrapolation of the total energy per atom. In this case, $E^{AB}(N_b)$, is known, and the model estimates the correction, $\Delta E^{AB}(N_b)$, the sum of both giving us the CBS estimate, $E^{AB}(N_\infty)$ [Eq. (1)]. Assessing the overall performance of our models, for both codes, we find that the QRF models achieve improved results in all metrics except for the 95% quantile AE for which the `exciting` model performs slightly worse. This indicates that for the largest CBS extrapolation corrections of the `exciting` dataset, which are around 20–40 eV/atom the stoichiometric

model is preferred. This knowledge gives us a better understanding of the QRF model's domain of applicability –i.e., the user should not expect more reliable CBS corrections than this from the stoichiometric model for calculated corrections of over 20 eV/atom. Overall, we understand that the improvement of the QRF models over the stoichiometric models comes from the addition of new features and the ability to fit nonlinear distributions.

The SISO model provides a small improvement in performance over the QRF model for `exciting` data. For FHI-aims data, however, the respective SISO model performs in between the stoichiometric model and the QRF model. In both SISO models, we find terms resembling the stoichiometric model; more weight is given though to the energy deviation of the solid of the more electronegative element. The similarity between the stoichiometric models and the SISO models may explain the discrepancy in performance for the two codes, since the stoichiometric model does well for `exciting` data but is quite poor for FHI-aims data where the SISO model also struggles. We recommend the QRF model over the SISO model since it performs well for both codes and provides the user with reliable prediction intervals. If the user, however, desires an interpretable model, the SISO model can be chosen.

The prediction intervals of the QRF offer the user a quantitative estimate of how uncertain the model is in each prediction it makes. We expect the model to have larger prediction intervals for larger targets and we see this with the 95% prediction intervals, which have over 0.69 Pearson correlation for both codes. The 95% prediction intervals contain slightly less than 95% of the held-out calculated CBS energy corrections. In general, however, the correlation of the prediction intervals and the percent of data they cover indicate the prediction intervals of the QRF models to be well behaved.

The QRF models offer users of materials databases a quantitative assessment of how far a total-energy result is from the CBS limit, which helps to evaluate how these results can be reused/repurposed. Looking forward, we believe that this work will allow such databases to provide estimated CBS

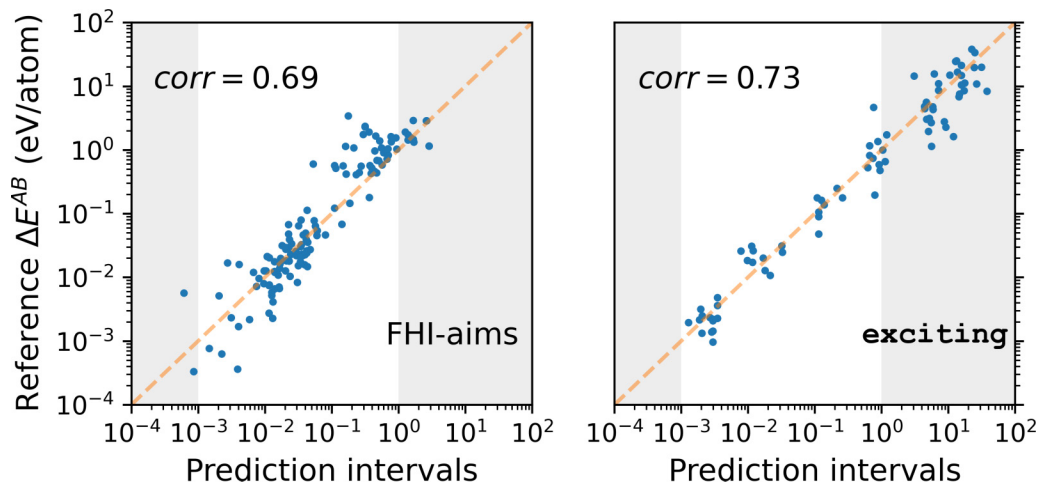


FIG. 5. The QRF model’s prediction interval size versus DFT CBS energy correction targets for FHI-aims (left) and *exciting* (right) data. Note, the logarithmic axes. This plot shows the effect of the calculated DFT targets on the associated prediction interval sizes of the QRF model. Larger/smaller ΔE^{AB} targets are generally associated with larger/smaller prediction intervals. The region of calculated ΔE^{AB} values between 1 meV/atom and 1 eV/atom are plotted with a lighter shade since it is of particular interest to DFT practitioners.

corrections for hosted data. We find the overall performance of the QRF models in terms of sMAPE on the test dataset of less than 25% for *exciting* and less than 15% for FHI-aims acceptable, especially considering that we provide prediction intervals that indicate the precision of the estimate (and thereby the domain of applicability of the model). This will help nonexperts unlock the large potential these databases have. For instance, data that were simulated for a molecular dynamics investigation, likely have a large CBS correction and might be unsuitable for investigations where high precision is required. On the opposite case, data performed with very high precision settings, even coming from different sources, will have a low CBS correction and might be suitable for a wide range of machine-learning tasks. We also believe that this work may have future applications in recommending a basis-set size to DFT practitioners for achieving a certain degree of precision before a calculation is performed, thereby saving computational expenses.

ACKNOWLEDGMENTS

This work received partial funding by the International Max Planck Research School for Elementary Processes in Physical Chemistry, the German Research Foundation (DFG) through the SFB 1404 (FONDA), Projektnummer 414984028, and the Max-Planck Graduate Center for Quantum Materials. The NOMAD Center of Excellence, Grant Agreement No. 951786 under the Union’s Horizon 2020 research and innovation program is appreciated. The authors gratefully acknowledge the computing time made available to them for the QRF hyperparameter search through The North-German Supercomputing Alliance (HLRN) project bep00098 on the high-performance computers Lise and Emmy at the NHR@ZIB and NHR@Göttingen respectively. These centers are jointly supported by the Federal Ministry of Education and Research and the state governments participating in the NHR (www.nhr-verein.de/unsere-partner).

D.S. performed the data analysis and wrote the quantile-random-forest analysis software. C.C. advised on the machine

learning and the FHI-aims calculations. L.G. advised on the machine learning. S.L. gave advice with respect to the *exciting* calculations. M.S. and C.D. designed the research questions and advised on all matters of the research. All authors contributed to the paper.

DATA AVAILABILITY

The raw DFT data, i.e., input and output files for both *exciting* and FHI-aims are hosted at NOMAD [32]. Note that these DOIs also contain other calculations that are not used in this paper.

The code used to generate all figures and train all models in this paper can be found on the NOMAD AI-Toolkit [27] in an online Jupyter notebook [34]. Parsing and feature engineering code can be found online [26].

APPENDIX

The stoichiometric models predictions on the test dataset are seen in Fig. 6. The performance of the SISSO models on the test dataset can be seen in Fig. 7.

1. SISSO dimensionality

The SISSO algorithm works by first choosing a dimensionality of the model. The model dimension is chosen by the performance of the different dimensionality models on the validation dataset. The validation dataset is chosen randomly as twenty-five percent of the training dataset. The performance of the models on the validation dataset as a function of dimension can be seen for both codes in Figure 8. We see that for FHI-aims a three dimensional model is optimal while for *exciting* we see negligible benefit in increasing the dimensionality from three to four. Therefore for both models we opt for a three dimensional model. The models presented in the paper are then retrained on the validation and training datasets combined.

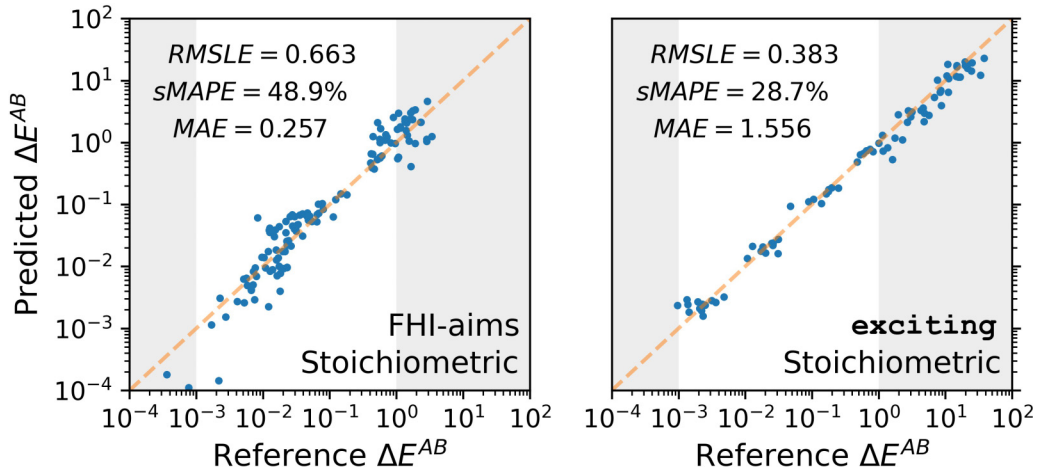


FIG. 6. Stoichiometric model predictions of ΔE^{AB} for FHI-aims (left) and **exciting** (right). The regions of particular interest to DFT practitioners, i.e., reference values between 1 meV/atom and 1 eV/atom, are highlighted by the white background.

2. Combination model

We also investigate the performance of a QRF model trained on the residuals (remaining error) of the stoichiometric model. In mathematical terms, the residual to which we fit is the variable: $\Delta E^{AB}(N_b) - C^A * \Delta E^A(N_b) - C^B * \Delta E^B(N_b)$. This means our estimate of the CBS total-energy correction is the sum of the stoichiometric model and a new QRF trained on the stoichiometric residuals. We call this model the combination model. The motivation is that subtracting the stoichiometric model from the target might allow the QRF to focus more on nonlinear contributions. The QRF model in the combination model is trained on the stoichiometric residuals. The stoichiometric model may under or overestimate the CBS corrections giving the residual positive and negative signs. As such we cannot use the RMSLE to optimize the combination model. Instead we need to optimize for a different metric, namely the symmetric mean absolute percentage error (sMAPE).

The resulting combination model returns a sMAPE of 26.4% and 14.8% for the **exciting** and FHI-aims codes, respectively. The MAPE is 27.2% and 21.5% for **exciting**

and FHI-aims respectively. The combination models perform worse than the QRF models in terms of test sMAPE and MAPE for both DFT codes. (Recall that the combination models are optimized for the sMAPE.) The MAE is 1.360 and 0.030 eV/atom for the **exciting** and FHI-aims codes respectively. This suggests the combination model does better than the QRF model for large targets. The hypothesis that the combination model outperforms the QRF model appears false over the wide range of targets but true for very large targets for **exciting**. This may be due to the fact the residuals of the stoichiometric model are in general quite large, e.g., the training sMAPE is 28.7% and 48.9% for **exciting** and FHI-aims respectively, which make training a QRF on these residuals too difficult a task. The FHI-aims combination model shares many of the very large errors of the stoichiometric model. This is expected since the former employs a QRF that is fit on the residuals of the latter. When these residuals are sporadically very large, as is the case for both codes containing SPEs from their respective stoichiometric models of close to 100% (see Fig. 2), the combination model has a difficult task to fit these residuals. We find the combination models to perform worse

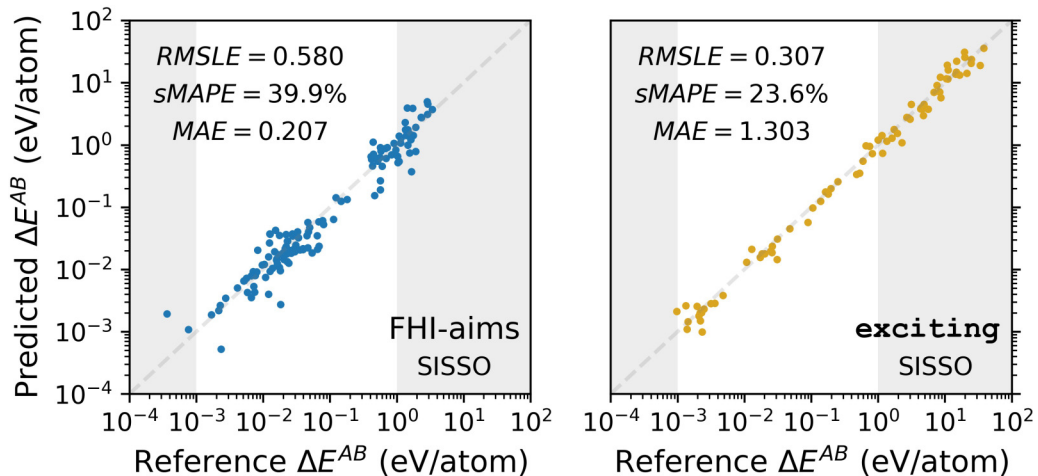


FIG. 7. SISSO three-dimensional model predictions of ΔE^{AB} for FHI-aims (left) and **exciting** (right). The regions of particular interest to DFT practitioners, i.e., reference values between 1 meV/atom and 1 eV/atom, are highlighted by the white background.

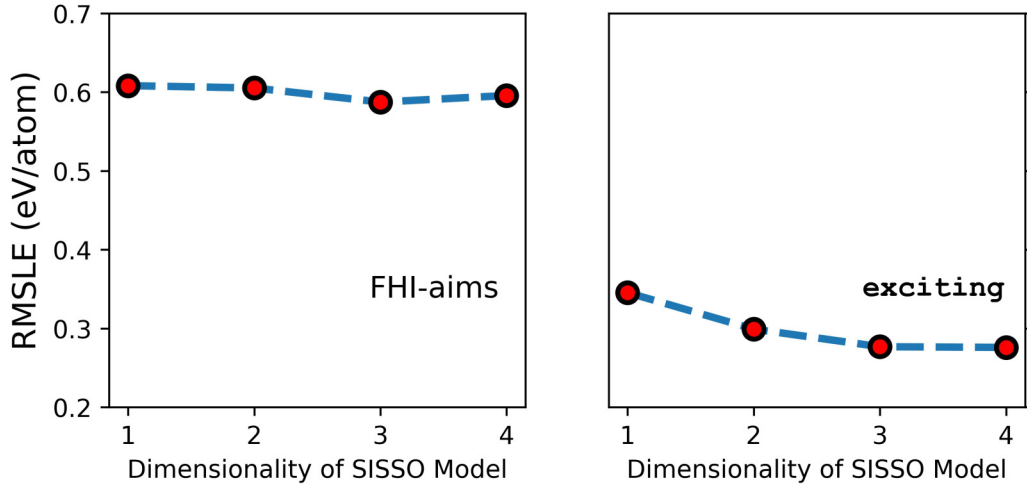


FIG. 8. SISSO model performance for FHI-aims (left) and *exciting* (right) on the validation datasets as a function of model dimension.

than the QRF models for FHI-aims in terms of our metrics that cover a wide range of targets (sMAPE, MAPE) and as such we do not recommend the use of the combination model.

3. Optimizing the RMSLE vs the RMSE

As already mentioned, optimizing for the RMSE is not recommended for targets that vary over several orders of magnitude. Since the square-root function is monotonic, optimizing the RMSLE is equivalent to optimizing the mean-squared error (MSE). For example, for a target of 1 meV per atom and prediction of 10 meV, we have a percentage error of 900%, a squared error of $0.081 \text{ (eV/atom)}^2$ and a squared logarithm error of 4.91. For a target of 1 eV per atom and a prediction of 1.5 eV per atom, the percentage error is 50%, the squared error is 0.25 eV per atom squared, and the log-squared error is 0.16. We can immediately see that the squared error is much larger for the larger target even though percentage wise, we are much closer to the target. This means that when optimizing for the RMSE, the larger targets are much more significant than the smaller ones that do not contribute very

much to the model's MSE/RMSE. This effect is demonstrated in Fig. 9 for the SISSO approach, where two models are shown, one trained on the RMSE and the other on the RMSLE. The model trained on the RMSE shows a good test data fit for large targets but a very poor fit for medium to smaller targets. The model approximates the latter with a constant estimate. The RMSLE-trained model, however, performs well across the orders of magnitude spanned by the data.

4. Decision tree optimization of RMSLE

Quantile random forests are built on decision trees, which typically are piecewise constant functions, defined as

$$h(x_i) = \sum_{m=1}^M c_m I(x_i \in R_m). \quad (\text{A1})$$

The index i refers to an individual input in the dataset identifying the compound, AB and basis-set size, N_b . The indicator function I is equal to one if the input data point x_i belongs to a region of the input data space R_m . The index m refers to the different regions of input space, which the decision

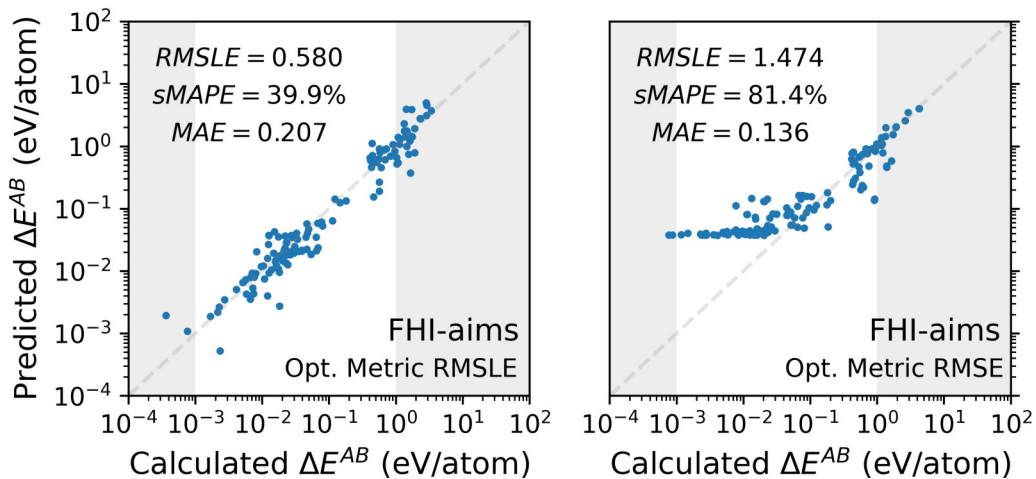


FIG. 9. SISSO model performance for FHI-aims using the RMSLE (left) and RMSE (right) metric for optimization. Metrics on the test dataset are also displayed for each resulting model.

tree partitions. These trees estimate a constant c_m when the input data point x_m falls into a region R_m , where m runs over the entire dataset. Note in general, and in the models used in this paper, the input features, x , are multidimensional (e.g., more than one one-dimensional feature is used for inference). The constants are chosen to minimize the metric of choice, in our case the RMSLE, for each region, R_m of input space. We minimize the mean-squared logarithmic error (MSLE) for simplicity since it is equivalent to minimizing the RMSLE since the square root function is convex. Our loss (cost) function, labeled J , is then defined as in Eq. (A2).

$$J = \sum_{i=1}^{i=N} \left\{ \log[y(x_i) + \epsilon] - \log \left[\sum_{m=1}^M c_m I(x_i \in R_m) \right] + \epsilon \right\}^2 \quad (\text{A2})$$

We take the derivative of the cost in this region with respect to the constant, $c_{m'}$, and set this derivative equal to zero. The constant, $c_{m'}$ for a region $R_{m'}$ is determined only by the input data that falls in that region. The model in Eq. (A1) predicts the constant $c_{m'}$ for data that falls in that region $R_{m'}$. We compare that prediction with the true value $y(x_i)$. To simplify our analysis, we focus on a single region $R_{m'}$ labeled with integer m' and set out to find the value of the constant $c_{m'}$ that minimizes the loss of that region, $J(c_{m'})$. The loss function thus becomes

$$J(c_{m'}) = \sum_{x_i \in R_{m'}} \{ \log[y(x_i) + \epsilon] - [\log(c_{m'} + \epsilon)] \}^2. \quad (\text{A3})$$

In this region, the only data points that affect the loss satisfy the condition $x \in R_{m'}$. Moreover, in this region we can replace our sum of indicator functions (piece-wise constant tree) defined in Eq. (A1) with a single constant, $c_{m'}$ for this region. We then arrive at Eq. (A4).

$$\frac{dJ(c_{m'})}{dc_{m'}} = -2 \frac{1}{(c_{m'} + \epsilon)} \sum_{x_i \in R_{m'}} \{ \log[y(x_i) + \epsilon] - \log(c_{m'} + \epsilon) \} = 0. \quad (\text{A4})$$

We note immediately that $c_{m'} = -\epsilon$, leaves our loss function gradient with a logarithmic argument of zero and a division by zero which is undefined and not a solution we desire. If $c_{m'} = -\epsilon$, we arrive at

$$\sum_{x_i \in R_{m'}} \{ \log[y(x_i) + \epsilon] \} = N_{m'} \log(c_{m'} + \epsilon) \quad (\text{A5})$$

where $N_{m'}$ is the number of data points in the region (i.e., satisfying $x_m \in R_{m'}$). The use of the RMSLE as a metric leads to

$$c_{m'} = \exp \left\{ \frac{1}{N_{m'}} \sum_{x_i \in R_{m'}} [\log(y(x_i) + \epsilon)] \right\} - \epsilon \quad (\text{A6})$$

as the choice of optimal constants for each region in our decision tree. Note, if we had chosen the MSE as a metric for the above analysis, we would have found the following constants, i.e., simply the average of targets in the region.

$$c_{m'} = \frac{1}{N} \sum_{x_i \in R_{m'}} y(x_i) \quad (\text{A7})$$

5. SISSO model predictions

As mentioned in the Sec. II Methods section, the SISSO API that we use does not have the RMSLE as an optimization metric. Therefore, we modify our target to perform an optimization that is analogous to optimizing the RMLSE [Eq. (7)]. The trained model outputs a prediction of $\log[\Delta E^A B(x_i) + \epsilon]$. To recover the desired target, $\Delta E^A B(x_i)$, we exponentiate the output of the model and subtract ϵ .

6. Cross validation of QRF models

For our QRF models, we optimize the number of decision trees in the forest, the fraction of features available at each split (often set to 0.33 as a default), and the minimum number of samples per split. We perform a grid search with the maximum fraction of features at each split to be in the set [0.1, 0.3, 0.6, 0.9, 1], the minimum samples per leaf ranges in the set [1, 3, 5], and the number of trees used in the random forest to be in the set [50, 75, 100, 125, 150, 200, 250, 300]. The best parameter combination is chosen via tenfold CV. For `exciting`, the best parameter combination is the fraction of features set to 0.9, the minimum number of samples per leaf to one, and the number of trees in the forest to 50. We immediately notice that the fraction of features is quite high at 0.9. This is because there are only three significant features in the `exciting` model as can be seen from the feature importance in Fig. 3, which indicates that the model struggles when making splits without these features. The minimum samples per leaf being one means that the regression trees are fully grown. This can often mean that the trees are subject to overfitting, but this does not show up in the cross validation or in a large discrepancy between training and test results. For FHI-aims, the best parameter combination differs only in the number of trees in the forest, which is 150. From the feature importance, the FHI-aims model also has only a few highly significant features, which is likely the cause of the high feature-fraction requirement. All validation RMSLE values for each parameter combination as well as the standard deviation of the validation RMSLE across the ten folds can be found in the accompanying Jupyter notebook.

We also added the band gap, $E_g^{AB}(N_b)$ into the FHI-aims QRF model as a feature. Its feature importance was 0.0076, which is close to a random variable (0.0013), and therefore the feature was dropped from subsequent model iterations.

7. Neural network optimization

A neural architecture grid search [33] was performed, searching over architectures with one and two hidden layers, respectively. For two hidden layers, the number of nodes in the second layer was always half of the number of nodes in the first layer. We searched over 16, 8, and 4 nodes in the first hidden layer. Dropout was added to avoid overfitting, we tried a rate of 0.05, 0.1, and 0.2. The learning rate was varied between 1E-5, 1E-4, and 1E-3 while the adam optimizer was used throughout the search. The batch size was either 64 or 32. The data was split into a 64% training data set, a 16% validation dataset and a 20% test dataset which was the same for other models. We chose the best architecture based on the validation dataset performance. We then retrained the best

model on the training dataset and validation dataset before evaluating on the test dataset. In general, we struggle with the training of the NN. The RMSLE is often undefined since the prediction of the neural network often goes to number below $1\text{E-}4$. To remedy this fact we tried optimizing for the RMSLE metric with epsilon switched to 1 instead of $1\text{E-}4$. We differ between the two metrics by referring to one as RMSLE+1 and the other as RMSLE+ $1\text{E-}4$ in the Jupyter Notebook. This helps with training but we still see sporadic undefined RMSLE numbers.

The best FHI-aims NN architecture has two hidden layers, 8 nodes in the first hidden layer and for in the second, a batch size of 32 a learning rate of $1\text{E-}3$ and a dropout rate of 0.05. The FHI-aims NN model achieves a training RMSLE+1 loss of 0.001 and MAE of 0.2022. However, during test we see a

much higher RMSLE+1 of 0.418 and MAE of 0.239. This points to the NN significantly overfitting the training data set which comes as no surprise given the size of the dataset. The test sMAPE is 125.14 which is more than twice as high as any of the other models reported in Table I.

The best exciting NN architecture had two hidden layers, 16 nodes in the first layer and 8 in the second layer. It uses a learning rate of $1\text{E-}5$, dropout of 0.05 and a batch size of 32. For the excitingNN model we see training and test metrics are quite poor. The RMSLE+ $1\text{E-}4$ is over 1 for both training and test. The sMAPE is over 100% for both training and test. It appears the training is not very stable despite attempts to fix this with varying the learning rate and dropout. As a result, we do not recommend this model for this task and dataset for either code.

-
- [1] J. G. Hill, K. A. Peterson, G. Knizia, and H.-J. Werner, Extrapolating MP2 and CCSD explicitly correlated correlation energies to the complete basis set limit with first and second row correlation consistent basis sets, *J. Chem. Phys.* **131**, 194105 (2009).
 - [2] D. Bakowies, Extrapolation of electron correlation energies to finite and complete basis set targets, *J. Chem. Phys.* **127**, 084105 (2007).
 - [3] P. Kraus, Basis set extrapolations for density functional theory, *J. Chem. Theory Comput.* **16**, 5712 (2020).
 - [4] P. Kraus, Extrapolating DFT toward the complete basis set limit: Lessons from the PBE family of functionals, *J. Chem. Theory Comput.* **17**, 5651 (2021).
 - [5] J. P. Perdew, K. Burke, and M. Ernzerhof, Generalized gradient approximation made simple, *Phys. Rev. Lett.* **77**, 3865, (1996).
 - [6] C. Draxl, and M. Scheffler, The nomad laboratory: From data sharing to artificial intelligence, *J. Phys.: Mater.* **2**, 036001 (2019).
 - [7] C. W. Andersen, R. Armiento, E. Blokhin, G. J. Conduit, S. Dwaraknath, M. L. Evans, Á. Fekete, A. Gopakumar, S. Gražulis, A. Merkys, F. Mohamed, C. Oses, G. Pizzi, G.-M. Rignanese, M. Scheidgen, L. Talirz, C. Toher, D. Winston, R. Aversa, K. Choudhary *et al.*, OPTIMADE, an API for exchanging materials data, *Sci. Data* **8**, 217 (2021).
 - [8] C. Carbogno, K. S. Thygesen, B. Bieniek, C. Draxl, L. M. Ghiringhelli, A. Gulans, O. T. Hofmann, K. W. Jacobsen, S. Lubeck, J. J. Mortensen, M. Strange, E. Wruss, and M. Scheffler, Numerical quality control for DFT-based materials databases, *npj Comput. Mater.* **8**, 69 (2022).
 - [9] K. Lejaeghere, G. Bihlmayer, T. Björkman, P. Blaha, S. Blügel, V. Blum, D. Caliste, I. E. Castelli, S. J. Clark, A. D. Corso, S. de Gironcoli, T. Deutsch, J. K. Dewhurst, I. Di Marco, C. Draxl, M. Duřak, O. Eriksson, J. A. Flores-Livas, K. F. Garrity, L. Genovese *et al.*, Reproducibility in density functional theory calculations of solids, *Science* **351**, aad3000 (2016).
 - [10] R. A. Shaw, The completeness properties of Gaussian-type orbitals in quantum chemistry, *Int. J. Quantum Chem.* **120**, e26264 (2020).
 - [11] E. Bosoni, L. Beal, M. Bercx, P. Blaha, S. Blügel, J. Bröder, M. Callsen, S. Cottenier, A. Degomme, V. Dikan, K. Eimre, E. Flage-Larsen, M. Fornari, A. Garcia, L. Genovese, M. Giantomassi, S. P. Huber, H. Janssen, G. Kastlunger, M. Krack *et al.*, How to verify the precision of density-functional-theory implementations via reproducible and universal workflows, *Nat. Rev. Phys.* **6**, 45 (2024).
 - [12] M. Bogojeski, L. Vogt-Maranto, M. E. Tuckerman, K.-R. Müller, and K. Burke, Quantum chemical accuracy from density functional approximations via machine learning, *Nat. Commun.* **11**, 5223 (2020).
 - [13] L. M. Ghiringhelli, J. Vybiral, S. V. Levchenko, C. Draxl, and M. Scheffler, Big data of materials science: Critical role of the descriptor, *Phys. Rev. Lett.* **114**, 105503 (2015).
 - [14] S. Makridakis, Accuracy measures: Theoretical and practical concerns, *Int. J. Forecasting* **9**, 527 (1993).
 - [15] S. Jachner, K. G. Van Den Boogaart, and T. Petzoldt, Statistical methods for the qualitative assessment of dynamic models with time delay (R package qualV), *J. Stat. Software* **22**, 1 (2007).
 - [16] S. Jaganathan and P. Prakash, A combination-based forecasting method for the M4-competition, *Int. J. Forecasting* **36**, 98 (2020).
 - [17] N. Meinshausen, and G. Ridgeway, Quantile regression forests, *J. Mach. Learn. Res.* **7**, 983 (2006).
 - [18] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning* (Springer, New York, 2013), Vol. 112, p. 18.
 - [19] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer, Berlin, 2009), Vol. 2.
 - [20] L. Breiman, Random forests, *Mach. Learn.* **45**, 5 (2001).
 - [21] D. A. Freedman, Bootstrapping regression models, *Ann. Stat.* **9**, 1218 (1981).
 - [22] R. Ouyang, S. Curtarolo, E. Ahmetcik, M. Scheffler, and L. M. Ghiringhelli, SISSO: A compressed-sensing method for identifying the best low-dimensional descriptor in an immensity of offered candidates, *Phys. Rev. Mater.* **2**, 083802 (2018).
 - [23] R. Ouyang, E. Ahmetcik, C. Carbogno, M. Scheffler, and L. M. Ghiringhelli, Simultaneous learning of several materials properties from incomplete databases with multi-task SISSO, *J. Phys.: Mater.* **2**, 024002 (2019).
 - [24] T. A. R. Purcell, M. Scheffler, C. Carbogno, and L. M. Ghiringhelli, SISSO++: A C++ implementation of the

- sure-independence screening and sparsifying operator approach, *J. Open Source Software* **7**, 3960 (2022).
- [25] T. A. Purcell, M. Scheffler, and L. M. Ghiringhelli, Recent advances in the SISSO method and their implementation in the SISSO++ code, *J. Chem. Phys.* **159**, 114110 (2023).
- [26] <https://github.com/speckhard/cbs-extrapolation>.
- [27] L. Sbailò, Á Fekete, L. M. Ghiringhelli, and M. Scheffler, The NOMAD artificial-intelligence toolkit: Turning materials-science data into knowledge and understanding, *npj Comput. Mater.* **8**, 250 (2022).
- [28] J. L. Hintze and R. D. Nelson, Violin plots: A box plot-density trace synergism, *Am. Stat.* **52**, 181 (1998).
- [29] D. Jha, L. Ward, A. Paul, W.-K. Liao, A. Choudhary, C. Wolverton, and A. Agrawal, *ElemNet*: Deep learning the chemistry of materials from only elemental composition, *Sci. Rep.* **8**, 17593 (2018).
- [30] T. Bechtel, D. T. Speckhard, J. Godwin, and C. Draxl, [arXiv:2309.06348](https://arxiv.org/abs/2309.06348).
- [31] T. Heskes, Practical confidence and prediction intervals, in *Advances in Neural Information Processing Systems*, edited by M. C. Mozer, M. Jordan, and T. Petsche (MIT Press, 1996), Vol. 9.
- [32] Exciting: DOI:[10.17172/NOMAD/2020.07.15-1](https://doi.org/10.17172/NOMAD/2020.07.15-1), FHI-aims: DOI:[10.17172/NOMAD/2020.07.27-1](https://doi.org/10.17172/NOMAD/2020.07.27-1).
- [33] D. T. Speckhard, K. Misiunas, S. Perel, T. Zhu, S. Carlile, and M. Slaney, Neural architecture search for energy-efficient always-on audio machine learning, *Neural Comput. Appl.* **35**, 12133 (2023).
- [34] <https://nomad-lab.eu/aitutorials/error-estimates-qrf>.