Hierarchical machine learning of potential energy surfaces

Pavlo O. Dral [iD], Alec Owens [iD], Alexey Dral [iD], and Gábor Csányi

COLLECTIONS

Paper published as part of the special topic on Collection and Machine Learning Meets Chemical Physics
Note: This paper is part of the JCP Special Topic on Machine Learning Meets Chemical Physics.

View Online   Export Citation   CrossMark

ARTICLES YOU MAY BE INTERESTED IN

DFTB+, a software package for efficient approximate density functional theory based atomistic simulations
The Journal of Chemical Physics **152**, 124101 (2020); https://doi.org/10.1063/1.5143190

PSI4 1.4: Open-source software for high-throughput quantum chemistry
The Journal of Chemical Physics **152**, 184108 (2020); https://doi.org/10.1063/5.0006002

Machine learning for interatomic potential models
The Journal of Chemical Physics **152**, 050902 (2020); https://doi.org/10.1063/1.5126336

Hierarchical machine learning of potential energy surfaces

View Online   Export Citation   CrossMark

Pavlo O. Dral,[1,a)] (ID) Alec Owens,[2] (ID) Alexey Dral,[3] (ID) and Gábor Csányi[4,b)]

AFFILIATIONS

[1] State Key Laboratory of Physical Chemistry of Solid Surfaces, Fujian Provincial Key Laboratory of Theoretical
and Computational Chemistry, Department of Chemistry, and College of Chemistry and Chemical Engineering,
Xiamen University, Xiamen 361005, China
[2] Department of Physics and Astronomy, University College London, Gower Street, WC1E 6BT London, United Kingdom
[3] BigData Team, 1A Tormoznoye Shosse Off 17, Yaroslavl, Yaroslavl 150022, Russian Federation
[4] Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, United Kingdom

Note: This paper is part of the JCP Special Topic on Machine Learning Meets Chemical Physics.
a) Author to whom correspondence should be addressed: dral@xmu.edu.cn. URL: http://dr-dral.com
b) Electronic mail: gc121@cam.ac.uk

ABSTRACT

We present hierarchical machine learning (hML) of highly accurate potential energy surfaces (PESs). Our scheme is based on adding predictions of multiple Δ-machine learning models trained on energies and energy corrections calculated with a hierarchy of quantum chemical methods. Our (semi-)automatic procedure determines the optimal training set size and composition of each constituent machine learning model, simultaneously minimizing the computational effort necessary to achieve the required accuracy of the hML PES. Machine learning models are built using kernel ridge regression, and training points are selected with structure-based sampling. As an illustrative example, hML is applied to a high-level *ab initio* CH_3Cl PES and is shown to significantly reduce the computational cost of generating the PES by a factor of 100 while retaining similar levels of accuracy (errors of ~1 cm^{-1}).

Published under license by AIP Publishing. https://doi.org/10.1063/5.0006498

I. INTRODUCTION

Detailed knowledge of the potential energy surface (PES) of an atomistic system is crucial for understanding its physicochemical behavior, e.g., its spectroscopic, thermochemical, and dynamical properties. Quantum chemistry (QC) provides a systematic route for constructing accurate representations of the PES, but in practice, the quality of the PES is strongly limited by the system size, which largely dictates the choice of electronic structure method and basis set. As the number of atoms increases, the required number of single point calculations to build the PES rapidly grows. The situation is also computationally challenging for small molecules when constructing highly accurate PESs, for example, in high-resolution molecular spectroscopy,[1] which requires rotation–vibration spectra to be calculated with sub-wavenumber accuracy (errors less than 1 cm^{-1}, 0.03 kcal/mol, $1.2 \cdot 10^{-4}$ eV, 12 J/mol, or 4.6 μHartree).

Such high requirements on the accuracy of the PES impose very strict standards on the quality of the QC calculations and on the density and coverage of grid points—nuclear geometries—on the PES. Wavefunction-based methods such as coupled cluster, usually coupled cluster singles, doubles, and perturbative triples [CCSD(T)], or multireference configuration interaction (MRCI) are employed along with large one-particle basis sets near the complete basis set (CBS) limit. Furthermore, the treatment of additional, smaller energy corrections to account for effects such as scalar relativity can recover more of the electron correlation energy[2,3] and greatly improve the PES. Naturally, calculations for each point can take many central processing unit (CPU) hours and amount to years of CPU-time for the entire PES. For example, a highly accurate *ab initio* PES of methane[4] required ten separate calculations at each point on a grid of 97 721 nuclear geometries, so nearly 1×10^6 single point calculations.

With recent advances in machine learning (ML) in QC,[5–8] there is scope for massive savings in computational effort, particularly with regard to constructing PESs.[9–11] QC can be used to calculate energies (and energy gradients) for only a small fraction of grid points, which are subsequently employed as the training set for an ML model capable of predicting the energy (and gradients) for any grid point of interest. Many ML methods have been suggested for representing PESs in different applications, e.g., in spectroscopy and (molecular) dynamics,[11–80] and in most published studies, ML is applied directly to learn the energies at the target QC level. However, it is known that learning energy differences between high- and low-level QC methods is more straightforward and requires far fewer training points,[80–102] as has been demonstrated, e.g., in our previous studies on the water PES[81] and on Δ-machine learning (Δ-ML).[82] It was also shown that combining several Δ-ML models achieves better performance and lowers the computational cost of generating the training data.[83,84] This fact has yet to be fully exploited in the construction of molecular ML PESs, and to the best of our knowledge, no such procedure has been devised to find the optimal training data. Furthermore, practical research usually requires knowledge about the choice of QC levels of theory, training set geometries, and sizes *before* generating the computationally intensive reference data.

In this study, we exploit the hierarchy of increasingly accurate but computationally costly quantum chemical methods to significantly reduce the computational cost of generating training data for accurate ML PESs. We introduce hierarchical ML (hML) for constructing the molecular PES and utilize a hierarchy of Δ-ML models trained on energies and energy corrections calculated with multiple QC methods. The training set size and composition of each of these models are optimized in a semi-automatic way *before* generating the computationally intensive QC reference data, chosen using structure-based sampling. Each constituent ML model is trained within the kernel ridge regression (KRR) approach. Our scheme is applied to a highly accurate *ab initio* CH_3Cl PES[103] with the aim of reproducing it as accurately as possible with the minimum amount of computational cost required to generate the reference energies and energy corrections for training.

II. METHODS

In this section, we provide a general description of the hML approach, discuss the dataset we tested hML on, and describe the details of how hML is applied to this dataset.

A. Hierarchical machine learning

In hierarchical machine learning (hML), the target property $\hat{y}_{\Sigma,i}$ for a point (geometry) i is calculated as the sum over the predictions $\hat{y}_{M,i}$ made by its constituent Δ-ML models,

$$\hat{y}_{\Sigma,i} = \sum_M \hat{y}_{M,i}(N_{\text{tr},M}). \tag{1}$$

Each Δ-ML model M may be trained on a different set of points, i.e., $\hat{y}_{M,i}$ is a function of both the training set composition and the training set size $N_{\text{tr},M}$. We designed the procedure to determine the optimal number of training points for each Δ-ML model to achieve the required accuracy of the hML predictions with as short as possible CPU-time t_Σ, needed to generate the reference data for training

all Δ-ML models. To do this, we minimize the following objective function for a set of training set sizes $\{N_{\text{tr},M}\}$:

$$\frac{e_\Sigma}{e_{\Sigma,\text{th}}} + \frac{t_\Sigma / t_{\Sigma,\text{max}}}{1 - s_{\text{th}}} + \frac{p}{e_{\Sigma,\text{th}}}, \tag{2}$$

where e_Σ and $e_{\Sigma,\text{th}}$ are the estimated and user-defined "target" errors of the hML model, respectively, $t_{\Sigma,\text{max}}$ is the CPU-time that would be necessary to generate the "pure QC PES," s_{th} is the user-defined "target time saving," and p is the penalty for too small training set sizes. Below, we describe all these terms.

The crucial problem is how to estimate the error of the hML model e_Σ for a given set of training set sizes $\{N_{\text{tr},M}\}$ *without* training all of the component Δ-ML models, because we want to circumvent the need to generate large amounts of costly QC reference data. To solve this problem, we estimate the hML error e_Σ on a very small test set of $N_{\text{rnd}} = 100$ points randomly drawn from the entire dataset. e_Σ is a function of both the reference values $\{y_{\Sigma,i}\}_{i=1}^{N_{\text{rnd}}}$ and the values $\{\hat{y}_{\Sigma,i}\}_{i=1}^{N_{\text{rnd}}}$ predicted by the hML model. The reference data $\{y_{\Sigma,i}\}_{i=1}^{N_{\text{rnd}}}$ are calculated with QC methods, but the predicted values are unknown without training the constituent Δ-ML models. Thus, we need to estimate the values $\{\hat{y}'_{M,i}(N_{\text{tr},M})\}_{i=1}^{N_{\text{rnd}}}$ to be predicted by the Δ-ML models. For the estimation, we make an assumption that for the *same training points*, the ratio between predictions made by the constituent Δ-ML models for each point i remains roughly the same and equal to the ratio calculated from the reference data. For the ratio between the Δ-ML model M and $M = 1$ and for $N_{\text{tr},1} = N_{\text{tr},M}$,

$$\frac{\hat{y}_{M,i}(N_{\text{tr},M})}{\hat{y}_{1,i}(N_{\text{tr},M})} \approx r_{M/1,i} = \frac{y_{M,i}}{y_{1,i}}. \tag{3}$$

We investigate the validity of this assumption in Sec. III while here, we focus on the details of the implementation.

From Eq. (3), we can estimate the value predicted by the Δ-ML model M trained on $N_{\text{tr},M}$ points as

$$\hat{y}'_{M,i}(N_{\text{tr},M}) \approx \hat{y}_{1,i}(N_{\text{tr},M})r_{M/1,i}. \tag{4}$$

The problem, therefore, reduces to making predictions $\hat{y}_{1,i}(N_{\text{tr},M})$ for each point i in the test set using the Δ-ML model $M = 1$ trained on $N_{\text{tr},M}$ points. It is solved by calculating reference data $\{y_{1,i}\}$ with the most computationally affordable QC method(s) and training the Δ-ML model $M = 1$ on an increasing number of training points. Practically, this is done with a small step for small training set sizes and with larger steps for larger training set sizes forming a set of the numbers of training points with the corresponding predicted values $\{(N_{\text{tr},1,j}, \{\hat{y}_{1,i}(N_{\text{tr},1,j})\}_{i=1}^{N_{\text{rnd}}})\}$. For $N_{\text{tr},1}$ between $N_{\text{tr},1,j}$ and $N_{\text{tr},1,j+1}$, we use a two-point linear interpolation to calculate the value of $\hat{y}_{1,i}(N_{\text{tr},1})$.

In our assumption, we also stipulated that the same training points, i.e., geometries in the training set, should be used for Eq. (3) to be valid. This is ensured by ordering the dataset points and then by pulling the first $N_{\text{tr},M}$ points into the training set for each Δ-ML model M. The geometries in the smaller training set always form a subset of any larger training set of any other Δ-ML model.

Alongside error estimate, the CPU-time t_Σ needed to generate the training data for all the Δ-ML models is of highest importance. Each Δ-ML model M is trained on $N_{\text{tr},M}$ differences between values calculated with higher- and lower-level QC approximations.

The number of required calculations with both QC approximations for this Δ-ML model is equal to $N_{tr,M}$. The same method can be used either as a high- or a low-level QC approximation for generating reference data for several Δ-ML models. Hence, we have a list with varying numbers of required calculations for each QC approximation. Since we pull training points from an ordered dataset as described above, the number of total calculations N_A with the QC approximation A is just the maximum number in that list. Now, we can estimate the CPU-time t_Σ from estimates of the CPU-time t_A needed to perform a QC calculation for a single point (geometry),

$$t_\Sigma = \sum_A N_A t_A. \tag{5}$$

We also use Eq. (5) to calculate the $t_{\Sigma,max}$ by setting all N_A to the largest number of training points $max(\{N_{tr,1,j}\})$ used to train the first Δ-ML model $M = 1$. It can be interpreted as the maximum CPU-time that would be necessary to spend in order to calculate the energies and energy corrections at all required QC approximations without any ML to generate the "pure QC PES" defined by a grid of points used to train the largest Δ-ML model $M = 1$. This is a very strict and technical definition, because in practice, hML models can be used for many more energy evaluations, e.g., in molecular dynamics simulations.

Now, we can calculate the first two terms of the objective function given by Eq. (2). If the optimization is performed using only these two terms, it can lead to very small training set sizes to reduce the estimated CPU-time, especially when very computationally intensive high-level QC approximations are involved. For very small training set sizes, however, ML predictions become increasingly unreliable. The validity of our approximation for the ratio [Eq. (3)] may deteriorate. Hence, the estimated error e_Σ can be overoptimistically too small due to the overestimated error compensation. Thus, we introduce an additional term p to penalize for very small training set sizes. This penalty is calculated as the sum of the estimated errors e_M of each Δ-ML model M,

$$p = \sum_M e_M \left(\{y_{M,i}\}_{i=1}^{N_{rnd}}, \{\hat{y}'_{M,i}(N_{tr,M})\}_{i=1}^{N_{rnd}} \right). \tag{6}$$

e_M are calculated in the same way as e_Σ. This penalty exploits the fact that e_Σ is usually not equal to $\sum_M e_M$, e.g., if e is the root-mean-squared error (RMSE), $\sum_M e_M$, therefore, decouples different Δ-ML models avoiding fortuitous error compensation.

Above, we described how to calculate e_Σ, t_Σ, and p entering Eq. (2). The user needs to choose parameters $e_{\Sigma,th}$ ("target error" of the hML model) and s_{th} ("target time saving"). They enter the objective function in order to scale the estimated error, penalty, and CPU-time saving. These parameters have intuitive meanings given by their names, but it is more appropriate to consider them as weights enabling the user to tune the optimization procedure as discussed below. We optimize the objective function in this study using the global optimization algorithm dual annealing[104–109] as implemented in SciPy.[110,111] This algorithm worked much better in our tests for many Δ-ML models than the Nelder–Mead simplex[112,113] or the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithms.[114]

After the training sets for each Δ-ML model are determined, the reference data are generated for them with the QC approximations

and the Δ-ML models trained. A better error estimate of the resulting hML model can then be made on the test set as it will not rely on the above assumption and will use the actual values $\{\hat{y}_{M,i}(N_{tr,M})\}_{i=1}^{N_{rnd}}$ instead of the estimated values $\{\hat{y}'_{M,i}(N_{tr,M})\}_{i=1}^{N_{rnd}}$. If the accuracy of this hML model is not sufficient, then the optimization procedure can be repeated with a lower value of the target error $e_{\Sigma,th}$, and additional reference data can be generated.

B. Dataset: *Ab initio* CH_3Cl potential energy surface

We use the dataset of 44 819 grid points of the published CBS-35^{HL} PES of CH_3Cl,[103] for which very accurate *ab initio* energies are available. The grid points represent deformed nuclear geometries of CH_3Cl with energies up to $hc \cdot 50\,000$ cm^{-1}, where c is the speed of light and h is the Planck constant. The *ab initio* energies, which we denote as TBE (theoretical best estimate), were calculated using a range of high-level electronic structure methods with large correlation consistent basis sets, namely, explicitly correlated CCSD(T)-F12b[115] coupled cluster calculations with extrapolation to the complete basis set (CBS) limit using the cc-pVTZ-F12 and cc-pVQZ-F12 basis sets,[116] plus additional energy corrections to account for core–valence (CV) electron correlation, higher-order (HO) coupled cluster terms beyond perturbative triples, scalar relativistic (SR) effects, and diagonal Born–Oppenheimer corrections (DBOC). In the following, we denote CCSD(T)-F12b with a complete basis set, cc-pVTZ-F12, and cc-pVQZ-F12 simply as CBS, VTZ, and VQZ, respectively. Full details of the *ab initio* calculations and grid generation can be found in the original work,[103] which was motivated by the need for a highly accurate rotation–vibration spectrum of methyl chloride.[117]

Here, we also report second-order Møller–Plesset perturbation theory (MP2) calculations with the augmented correlation consistent basis set, aug-cc-pVQZ,[118,119] on the same grid of 44 819 geometries. MP2 calculations are computationally cheap and often used for initial explorations of molecular PESs. Calculations were performed using MOLPRO2012.[120] In the following, this dataset is referred to simply as MP2.

In Table I, we report the computational cost of calculating the required energies and energy corrections in the dataset. Note that the CBS is the composite method requiring calculations at both VTZ and VQZ for extrapolation. Since this extrapolation does not incur any noticeable additional cost, it is set to zero. We also estimate the influence of the level of theory used for the energy calculations with respect to TBE. For this, we use the following unweighted RMSE and weighted RMSE (wRMSE) for the energies calculated with the QC approximation A:

$$RMSE = \sqrt{\frac{1}{N} \sum_i^N (E_{A,i} - E_{TBE,i})^2}, \tag{7}$$

$$wRMSE = \sqrt{\frac{1}{N} \sum_i^N (E_{A,i} - E_i^{TBE})^2 w(E_{TBE,i})}. \tag{8}$$

The weights $w(E)$ reflect the importance of the more spectroscopically relevant low-energy region of the PES close to equilibrium

TABLE I. Computational cost of calculations on an Intel Xeon E5-2690 v2 3.0 GHz processor at the equilibrium geometry of CH_3Cl (CPU-seconds and % with respect to TBE). CPU-years for the entire grid are estimated as CPU-seconds multiplied by 44 819. Errors of the energies calculated at different levels vs TBE (RMSE, wRMSE, and wllRMSE) for the entire CH_3Cl PES are in cm^{-1}. Standard deviations (SD, wSD, and wllSD) for each energy correction from the zero baseline for the entire PES are in cm^{-1}.[a]

Method	CPU-seconds per grid point	CPU-years for entire grid	% of CPU-time with respect to TBE	RMSE/SD	wRMSE/wSD	wllRMSE/wllSD
MP2	296.85	0.42	0.31	746.41	202.26	183.30
VTZ	298.24	0.42	0.31	108.46	55.44	54.61
VQZ	2 224.71	3.16	2.31	95.40	48.75	48.10
CBS	0.00	0.00	0.00	93.01	47.42	46.78
SR	2 059.50	2.92	2.14	18.84	8.37	7.96
CV	12 450.03	17.68	12.95	129.37	63.64	61.66
HO	49 711.76	70.60	51.71	62.43	21.72	19.74
DBOC	29 400.00	41.75	30.58	6.18	3.11	3.04
TBE[b]	96 144.24	136.55	100.00	0.00	0.00	0.00

[a]For energies (MP2, VTZ, VQZ, and CBS) calculated as the error of the energy at each respective level vs TBE. For energy corrections CV, HO, SR, and DBOC calculated as the error of the TBE energy minus the value of the energy correction vs TBE.
[b]TBE does not include MP2. It also does not include the computational cost for extrapolation in the CBS (see the text).

and are calculated as described in previous studies using cm^{-1} units throughout,[12,103,121]

$$w(E) = \left(\frac{\tanh[-0.0006 \times (E - 15\,000)] + 1.002\,002\,002}{2.002\,002\,002} \right) \times \frac{1}{0.0001 \max(E, 10\,000)}. \quad (9)$$

For a QC correction A (DBOC, SR, CV, or HO), we also provide the standard deviation (SD) and weighted standard deviation (wSD) of the correction from the zero baseline,

$$SD = \sqrt{\frac{1}{N} \sum_i^N E_{A,i}^2}, \quad (10)$$

$$wSD = \sqrt{\frac{1}{N} \sum_i^N E_{A,i}^2 \, w(E_{TBE,i})}. \quad (11)$$

This definition of the standard deviation is equivalent to calculating the corresponding RMSEs for the TBE energies without a QC correction A, i.e., it shows how the PES quality would deteriorate if the QC correction was not included.

From Table I, it is clear that the most CPU-time is spent on calculating higher-level energies and energy corrections. The most approximate level of theory MP2 requires almost negligible amount of CPU-time for a rough estimate of the PES possessing a relatively small unweighted error of ~2 kcal/mol. This error of the MP2 PES is, nevertheless, unacceptably high for calculations of rovibrational spectra. We call MP2 the lowest level (*ll*) QC approximation.

C. Application of hierarchical machine learning to the CH3Cl potential energy surface

The goal of applying hML to the CH_3Cl PES is to reproduce the TBE energies ($y_{\Sigma,i} = E_i^{TBE}$) as accurately as possible with the least amount of CPU-time needed to generate the reference data for training. As discussed above, hML consists of multiple Δ-ML models. The hML model is constructed automatically given a hierarchy of QC approximations for energies (QC hierarchy), energy corrections (QC corrections), and defined composite methods. Below, we show an example of the dataset of the CH_3Cl PES on how all three components are used.

The QC hierarchy is an ordered list of QC approximations with increasing accuracy. In our case, the QC hierarchy is MP2, VTZ, VQZ, and CBS. We define Δ-ML models for learning the differences between each higher-level QC approximation on the right and the next lower-level QC approximation on the left. We add the baseline zero "0" in front of this list, so we have as many Δ-ML models as there are QC approximations in the QC hierarchy. This gives rise to the following four Δ-ML models:

1. Δ_0^{MP2} gives estimates $\hat{y}_{1,i}(N_{tr,1})$ after being trained on $N_{tr,1}$.
2. Δ_{MP2}^{VTZ} gives estimates $\hat{y}_{2,i}(N_{tr,2})$ after being trained on $N_{tr,2}$.
3. Δ_{VTZ}^{VQZ} gives estimates $\hat{y}_{3,i}(N_{tr,3})$ after being trained on $N_{tr,3}$.
4. Δ_{VQZ}^{CBS} gives estimates $\hat{y}_{4,i}(N_{tr,4})$ after being trained on $N_{tr,4}$.

Next, we define Δ-ML models for each correction, where order is not important. In our case, we have four more additional Δ-ML models:

5. Δ_0^{SR} gives estimates $\hat{y}_{5,i}(N_{tr,5})$ after being trained on $N_{tr,5}$.

6. Δ_0^{CV} gives estimates $\hat{y}_{6,i}(N_{\mathrm{tr},6})$ after being trained on $N_{\mathrm{tr},6}$.
7. Δ_0^{HO} gives estimates $\hat{y}_{7,i}(N_{\mathrm{tr},7})$ after being trained on $N_{\mathrm{tr},7}$.
8. Δ_0^{DBOC} gives estimates $\hat{y}_{8,i}(N_{\mathrm{tr},8})$ after being trained on $N_{\mathrm{tr},8}$.

Thus, the hML model for the CH_3Cl PES consists of eight Δ-ML models,

$$\mathrm{hML} = \Delta_0^{\mathrm{MP2}} + \Delta_{\mathrm{MP2}}^{\mathrm{VTZ}} + \Delta_{\mathrm{VTZ}}^{\mathrm{VQZ}} + \Delta_{\mathrm{VQZ}}^{\mathrm{CBS}} + \Delta_0^{\mathrm{SR}} + \Delta_0^{\mathrm{CV}} + \Delta_0^{\mathrm{HO}} + \Delta_0^{\mathrm{DBOC}}. \quad (12)$$

After all the Δ-ML models are defined, the procedure described above is used to determine their training set sizes $\{N_{\mathrm{tr},M}\}_{M=1}^8$. We start with ordering the grid points on the CH_3Cl PES using structure-based sampling[12,122] [related to farthest-point sampling (FPS) also used in constructing ML PESs][122–124] of all points from three slices (the procedure identical to the one described in the previous work[12]). In this procedure, the points farthest apart are selected iteratively, i.e., the ordering is ensured automatically. We then select one hundred random geometries for the test set with associated energies and energy corrections at all QC approximations from the dataset, from which we can calculate the ratios $r_{M/1,i}$. We train the Δ-ML model 1 (Δ_0^{MP2}) on an increasing number of training points in the interval [18, 44 817] taken from the beginning of the ordered grid points. Predictions are made with this Δ-ML model for each point in the test set, which allows us to estimate the error e_Σ and the penalty p terms for the optimization of the training set sizes.

To estimate the CPU-time needed to generate reference energies and energy corrections for given training sets, we take into account that the number of training points for the models based on composite QC methods, such as CBS, translates into the same number of required calculations of each constituent QC approximation used in the composite QC method. $t_{\Sigma,\mathrm{max}}$ is calculated as if each Δ-ML model was trained on the entire PES. We set the target time saving s_{th} to 0.99 (99%) in all optimizations.

In order to monitor the error of the hML model and its component Δ-ML models, we want to use the wRMSE measure for the CH_3Cl PES as was done in previous studies.[12,103] However, this requires knowledge of the TBE energies [see Eq. (8)], which are not always available, i.e., in cases where the number of training points is different. Thus, we suggest to use the weights $w(E_{ll,i})$ estimated with the lowest-level and computationally most affordable QM approximation (MP2) instead of $w(E_{\mathrm{TBE},i})$. This error estimate is denoted as wllRMSE to distinguish it from the true wRMSE. Analogously, wllSD can be defined. Using ll energies instead of TBE has a relatively small effect on the weighted RMSE and SD (Table I). We use wllRMSE as the error e both in the optimization procedure and to estimate the hML error after its constituent Δ-ML models are trained.

Each Δ-ML model is trained using the ML technique described in the previous study.[12] In short, we use the self-correcting kernel ridge regression (KRR) with the Gaussian kernel and the RE molecular descriptor given by a vector of normalized inverted internuclear distances. We use the four layers of the self-correcting KRR approach, where each next layer corrects the residual error of the previous layer. To account for the equivalence of the homonuclear atoms, a permutationally invariant (symmetrized) kernel[10,12,122] is used. All calculations were performed with MLatom,[125] and the implementation details are described elsewhere.[122] The only difference in this study is that in the hyperparameter optimization procedure, we use ll (MP2) energies rather than TBE energies as an

additional energy criterion (energies below 10 000 cm^{-1}) for selecting the validation set from the training set for the reasons explained above.

III. RESULTS AND DISCUSSION

In this section, we use the CH_3Cl PES to investigate the validity and limitations of the assumptions made and to demonstrate the application of hML. We note that in this study, we used the same sampling procedure (structure-based sampling) and trained each Δ-ML model using the same setup as that in our previous study[12] (with some slight modifications as described above). For a specific application, it is recommended to choose the best setup (sampling, ML algorithm, etc.) based on tests using the low-cost reference data generated with the lowest-level QC approach.

A. Validity of assumptions

Evaluating the error on the small test set of $N_{\mathrm{rnd}} = 100$ random points gives a faithful estimate of the actual error for the entire PES, as can be seen on an example of the learning curve for Δ_0^{MP2} (Fig. 1). The learning curve of the wRMSE for the entire PES of 44 819 grid points does not decrease monotonically. Nevertheless, the learning curve of the wllRMSE on the test set captures the correct shape of the learning curve of the wRMSE with all the peaks.

Now, we turn to the main assumption made in the optimization procedure, i.e., that the ratio between the predictions made by the constituent Δ-ML models trained on the same geometries for any new point is approximately constant [Eq. (3)]. The learning curves of the ratios $r_{M/1}$ of the predictions made by a selection of Δ-ML models and the Δ-ML model 1 (Δ_0^{MP2}) are plotted for several test points and compared with the actual ratios (Fig. 2). The assumption holds rather well, and the ratios practically lie on the lines of true ratios for large training sizes. This assumption only breaks down for very small training sets, especially with much fewer than
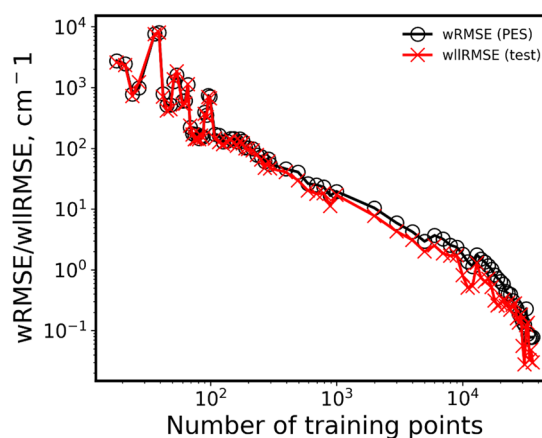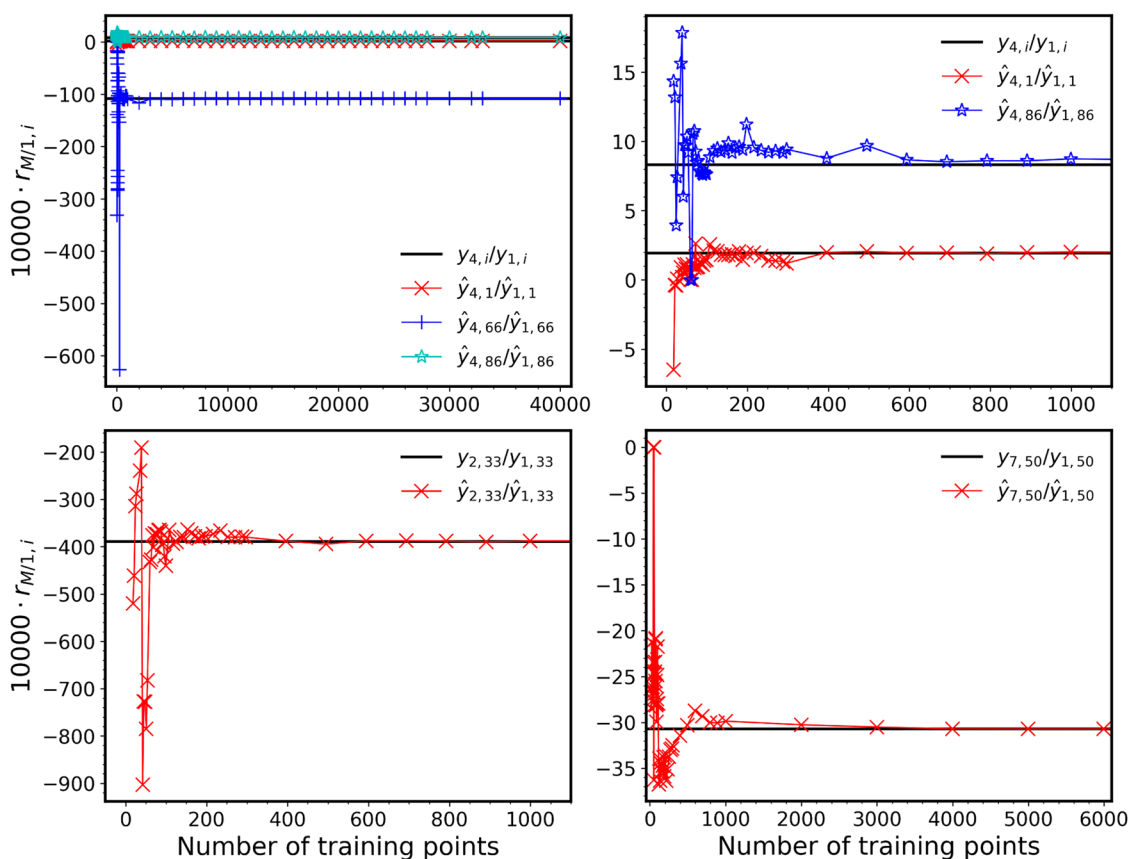

FIG. 1. The evolution of the wRMSE and wllRMSE of Δ_0^{MP2} with increasing training set size as evaluated on the test set of $N_{\mathrm{rnd}} = 100$ random points and the entire PES of 44 819 grid points, respectively. The plot is shown with log scaling for wRMSEs above 0.05 cm^{-1}.

FIG. 2. The evolution of the ratio between the predictions made by a selection of Δ-ML models [M = 4 ($\Delta_{\text{VQZ}}^{\text{CBS}}$), 2 ($\Delta_{\text{MP2}}^{\text{VTZ}}$), and 7 ($\Delta_0^{\text{HO}}$)] and the Δ-ML model 1 (Δ_0^{MP2}) with increasing training set size compared to the actual ratios for several random points (i = 1, 66, 86, 33, and 50). Note the different intervals on the x-axis and scales on the y-axis. Since the ratios are much smaller than one, they were multiplied by 10 000.

100 training points. This issue is addressed by including the penalty term in the optimization procedure.

We note that the ratios are usually much smaller than unity because the corrections to the rough energy estimate done with the *ll* QC method (in our case MP2) are much smaller than the energy estimate itself. Thus, the errors of the Δ-ML models trained on these corrections are also numerically much smaller and not very sensitive to moderate deviations of the ratios from their true values.

B. Generation and performance of hML models

We applied our optimization procedure to obtain the number of training points for each constituent Δ-ML model of the hML model for the CH_3Cl PES. We set the target error $e_{\Sigma,\text{th}}$ to 1.00 cm^{-1}. The optimized training set sizes of the resulting hML model 1 (hML1), its estimated error e_Σ, and penalty p are reported in Table II. After training the hML1 model on these training sets, the actual wllRMSE on the same test set as that used for optimization was calculated to be 2.23 cm^{-1}, i.e., significantly higher than the target and estimated errors. It is noteworthy that the actual wRMSE on the entire PES (2.44 cm^{-1}) is very close to the above wllRMSE,

i.e., wllRMSE evaluated on the small test set proved to be a reliable indicator of the real error.

To further improve the accuracy, we decreased the user-defined parameter $e_{\Sigma,\text{th}}$ to 0.50 cm^{-1}. This led to the second-iteration hML model (hML2, Table II), which already achieved a spectroscopic accuracy of 1 cm^{-1} with training sets that would require spending only 1% of the CPU-time needed for the pure *ab initio* PES. It is clear, however, that some of the training set sizes of hML1 are larger than those of hML2, i.e., more reference data are available after the first iteration than those used in the second iteration. This is because the optimization procedure can find multiple minima with similar values for the objective function, but with very different training set sizes. So, it may happen that fewer training points are needed for some Δ-ML models in the next iteration. This is balanced out by more training data for other Δ-ML models.

One of the approaches to avoid losing reference data available in the previous iteration is to train the hML model on the union of reference data from hML1 and hML2. Such an hML model (hML1∪2, Table II) is only nominally better than hML2. It also achieves spectroscopic accuracy while saving 99% of the CPU-time. Here, the user should be careful that the combined CPU-cost of

TABLE II. Hierarchical machine learning (hML) models for the CH_3Cl PES created in this study: training set sizes N_{tr} of the constituent Δ-ML models, saved time $s = 100(1 - t_\Sigma/t_{\Sigma,max})$ in %, wRMSEs and RMSEs evaluated on the entire PES, wllRMSEs evaluated on the test set, $e_{\Sigma,th}$, e_Σ, and p. All errors are in cm^{-1}.

	hML1	hML2	hML1∪2	hML3	hML4	10%-hML
Δ_0^{MP2}	30 727	30 720	30 727	30 735	41 985	4480
Δ_{MP2}^{VTZ}	5 261	8 433	8 433	7 595	12 135	4480
Δ_{VTZ}^{VQZ}	915	1 653	1 653	1 536	2 117	4480
Δ_{VQZ}^{CBS}	915	1 653	1 653	1 536	2 117	4480
Δ_0^{SR}	749	860	860	754	863	4480
Δ_0^{CV}	881	746	881	900	911	4480
Δ_0^{HO}	122	274	274	122	587	4480
Δ_0^{DBOC}	87	112	112	87	119	4480
s	99.22	99.00	98.96	99.26	98.46	90.00
RMSE	13.88	11.23	11.08	13.46	10.11	58.81
wRMSE	2.44	1.12	1.12	2.44	0.91	3.49
wllRMSE	2.23	1.01	1.00	2.19	0.85	2.72
e_Σ	0.18	0.15	0.18	0.18	0.10	2.77
p	0.44	0.39	0.36	0.42	0.23	2.86

study[12]), meaning that learning the energy components separately on the same set of training points gives the same results as those of learning the combined energy. Thus, in practical situations, for very similar training set sizes, it makes sense to combine several Δ-ML models into a single Δ-ML model in order to decrease the calculation time with ML for new points. This may become relevant in applications with many ML function calls, e.g., in long molecular dynamics, Monte Carlo simulations, or rovibrational calculations.

In the 10%-hML model, relatively large training sets of 4480 points were used. Consequently, our assumptions become more valid, and the error estimate $e_\Sigma = 2.77$ cm^{-1} made with these assumptions on 100 test points is almost the same as the actual wllRMSE = 2.72 cm^{-1} for the same test points, and very close to the wRMSE = 3.49 cm^{-1} for the entire PES. Another interesting observation is that the penalty $p = 2.86$ cm^{-1} is almost the same as e, while for small training set sizes, the penalty was twice as large as the estimated error.

generating the reference data for hML1 and hML2 is not too large. This can be tuned by re-running the optimization procedure in the second iteration with different parameters and calculating the combined cost before generating the training data.
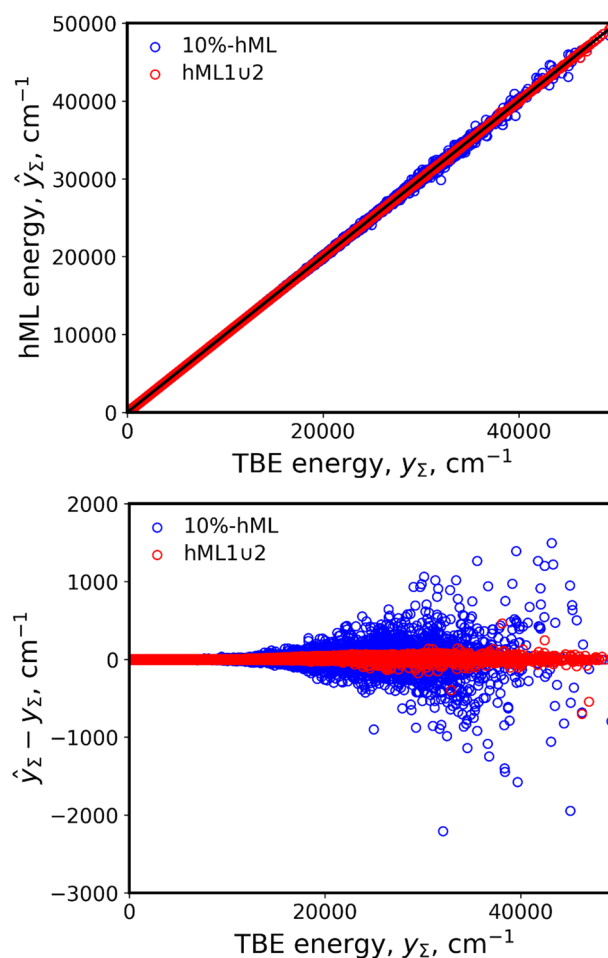
Another approach is to set the lower boundary for the training set sizes in the second optimization iteration to those obtained in the first optimization iteration. This would naturally solve the problem of losing a part of the reference data generated in the first iteration. We re-ran the second optimization iteration starting with the optimal number of training points from the hML1 model as the lower boundary and trained the hML3 model on the resulting training set sizes. We used the same parameter $e_{\Sigma,th} = 0.50$ cm^{-1} for optimizing the hML2 model. The estimated e_Σ and p for the hML3 model are not much better than those for hML1. Indeed, the trained hML3 model was no better than hML1 and much worse than hML2 (compare wRMSEs and RMSEs, Table II).

Apparently, the optimization algorithm that performed the search with more constraints on the training set sizes failed to find as good an hML composition as that in the hML2 case. Decreasing the parameter $e_{\Sigma,th}$ to 0.25 cm^{-1} and using the optimized training set sizes from hML3 as the lower boundary in the third optimization iteration led to the hML4 model. This model is very similar to hML1∪2 in terms of achieved accuracy and saved CPU-time (Table II). The main difference is that the achieved error is noticeably lower than 1 cm^{-1} (wRMSE of 0.91 cm^{-1} vs 1.12 cm^{-1}) with a slightly decreased computational time saving (98.5% vs 99%), while the unweighted RMSE is also lower (10.11 cm^{-1} vs 11.08 cm^{-1}).

For comparison, we created an hML model with all Δ-ML models trained on 10% of the PES grid points (Table II). This 10%-hML model is essentially equivalent to the s10%-ML model trained directly on TBE energies for the same 10% of points (see the previous

FIG. 3. Top: energies evaluated with QC (TBE) and the hML1∪2 and 10%-hML models. Bottom: errors of the hML1∪2 and 10%-hML models relative to TBE. All energies and errors are plotted for the entire CH_3Cl PES and all TBE energies.

Our hML1∪2 model saves ten times more CPU-time than the previously reported s10%-ML model[12] or the 10%-hML model generated here and at the same time has three times smaller wRMSE and five times smaller RMSE. The energies predicted by the hML1∪2 and 10%-hML models compared to the TBE energies are given in Fig. 3, which shows that better correlation and less outliers are achieved with hML1∪2. hML1∪2 is even better than the previously reported 25%-ML model trained on 11 204 points (wRMSE of 1.39 cm^{-1} and RMSE of 13.96 cm^{-1}, saving 75% CPU-time).[12] We also assessed the quality of the hML1∪2 model from a different perspective; the energies predicted by it for the entire CH_3Cl PES were used for calculating vibrational energy levels using the same procedure as that in the original studies.[12,103] The quality of the resulting hML1∪2 vibrational energy levels with respect to the pure *ab initio* levels[103] is comparable to the quality of vibrational energy levels obtained with previously reported s10%-ML and 25%-ML models[12] with an RMSE of only 0.17 cm^{-1} (see the supplementary material).

IV. CONCLUSIONS

In this study, we propose a hierarchical machine learning (hML) scheme composed of multiple Δ-ML[82] models for constructing highly accurate potential energy surfaces (PESs). This scheme exploits the hierarchy of increasingly accurate but computationally costly quantum chemical (QC) methods to generate reference energies and energy corrections for training the constituent Δ-ML models. Our scheme semi-automatically defines the composition of an hML model including the training set sizes of each constituent Δ-ML model *before* performing computationally expensive QC calculations. The number of training points is optimized to achieve the target accuracy of the hML PES with the minimum amount of CPU-time needed for generating reference energies and energy corrections with the QC approximations.

It is known that the number of computationally expensive high-level QC calculations can be greatly reduced by combining several Δ-ML models, some of which are trained on many more low-level QC data.[83,84] However, the choice of the optimal number of training points for each constituent Δ-ML model is not trivial, especially for a large number of the Δ-ML models. To the best of our knowledge, until now, no procedure was suggested to determine the training set sizes ahead of time. Here, we developed such an optimization procedure.

The training set sizes are determined by exploiting an assumption introduced here that the ratio of predictions between two Δ-ML models trained on the same geometries remains approximately constant. This allows us to estimate the Δ-ML model predictions for a given training set size before training the Δ-ML models, if the prediction of the Δ-ML model trained on the least computationally expensive QC energies is known. Thus, most of the CPU-cost incurred during the optimization procedure comes from generating reference energies at the lowest-level QC approximation, which is negligible in the case of high-accuracy PESs. Here, we have shown that this assumption works rather well and only breaks down for very small training set sizes.

During the optimization of training set sizes, the CPU-cost that would be necessary for generating reference energies and energy corrections is minimized together with the estimated error of the hML model. This error is estimated on a small test set (in this study 100 points). We have demonstrated that these error estimates are rather good for larger training set sizes, but become increasingly less accurate for very small training set sizes. However, the error of the hML model can be reliably determined after all constituent Δ-ML models are trained, and the training set sizes can be re-optimized in an iterative manner with stricter optimization criteria until the hML error is sufficiently low.

The numerical validation of our hML approach was performed on an example of CH_3Cl PES, for which highly accurate *ab initio* data at various QC levels were available. We have shown that the hML scheme can reproduce this *ab initio* PES with an error of ~1 cm^{-1}, while reducing the required amount of CPU-time for generating the QC energies and energy corrections by a factor of 100.

SUPPLEMENTARY MATERIAL

See the supplementary material for the analysis of the vibrational energy levels of CH_3Cl calculated using the potential energy surface obtained with the hML1∪2 model.

DATA AVAILABILITY

All datasets used in this study containing grid geometries, energies, and energy corrections calculated with the quantum chemical approximations are openly available in figshare at https://doi.org/10.6084/m9.figshare.11876364.[126]

REFERENCES

[1] J. Tennyson, "Perspective: Accurate ro-vibrational calculations on small molecules," J. Chem. Phys. **145**, 120901 (2016).

[2] T. Helgaker, W. Klopper, and D. P. Tew, "Quantitative quantum chemistry," Mol. Phys. **106**, 2107–2143 (2008).

[3] K. A. Peterson, D. Feller, and D. A. Dixon, "Chemical accuracy in *ab initio* thermochemistry and spectroscopy: Current strategies and future challenges," Theor. Chem. Acc. **131**, 1079 (2012).

[4] A. Owens, S. N. Yurchenko, A. Yachmenev, J. Tennyson, and W. Thiel, "A highly accurate *ab initio* potential energy surface for methane," J. Chem. Phys. **145**, 104305 (2016).

[5] P. O. Dral, "Quantum chemistry in the age of machine learning," J. Phys. Chem. Lett. **11**, 2336–2347 (2020).

[6] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, and A. Walsh, "Machine learning for molecular and materials science," Nature **559**, 547–555 (2018).

[7] O. A. von Lilienfeld, "Quantum machine learning in chemical compound space," Angew. Chem., Int. Ed. **57**, 4164–4169 (2018).

[8] M. Rupp, O. A. von Lilienfeld, and K. Burke, "Guest Editorial: Special topic on data-enabled theoretical chemistry," J. Chem. Phys. **148**, 241401 (2018).

[9] T. Mueller, A. Hernandez, and C. Wang, "Machine learning for interatomic potential models," J. Chem. Phys. **152**, 050902 (2020).

[10] A. P. Bartók and G. Csányi, "Gaussian approximation potentials: A brief tutorial introduction," Int. J. Quantum Chem. **115**, 1051–1057 (2015).

[11] J. Behler, "Perspective: Machine learning potentials for atomistic simulations," J. Chem. Phys. **145**, 170901 (2016).

[12] P. O. Dral, A. Owens, S. N. Yurchenko, and W. Thiel, "Structure-based sampling and self-correcting machine learning for accurate calculations of potential energy surfaces and vibrational levels," J. Chem. Phys. **146**, 244108 (2017).

[13] A. C. P. Bittencourt, F. V. Prudente, and J. D. M. Vianna, "The fitting of potential energy and transition moment functions using neural networks: Transition probabilities in OH ($A^2\Sigma^+ \rightarrow X^2\Pi$)," Chem. Phys. **297**, 153–161 (2004).

[14] S. Manzhos and T. Carrington, Jr., "A random-sampling high dimensional model representation neural network for building potential energy surfaces," J. Chem. Phys. **125**, 084109 (2006).

[15] S. Manzhos, X. G. Wang, R. Dawes, and T. Carrington, Jr., "A nested molecule-independent neural network approach for high-quality potential fits," J. Phys. Chem. A **110**, 5295–5304 (2006).

[16] F. V. Prudente, P. H. Acioli, and J. J. S. Neto, "The fitting of potential energy surfaces using neural networks: Application to the study of vibrational levels of H_3^+," J. Chem. Phys. **109**, 8801–8808 (1998).

[17] T. M. Rocha Filho, Z. T. Oliveira, L. A. C. Malbouisson, R. Gargano, and J. J. Soares Neto, "The use of neural networks for fitting potential energy surfaces: A comparative case study for the H_3^+ molecule," Int. J. Quantum Chem. **95**, 281–288 (2003).

[18] P. M. Agrawal, L. M. Raff, M. T. Hagan, and R. Komanduri, "Molecular dynamics investigations of the dissociation of SiO_2 on an *ab initio* potential energy surface obtained using neural network methods," J. Chem. Phys. **124**, 134306 (2006).

[19] N. Artrith, A. Urban, and G. Ceder, "Constructing first-principles phase diagrams of amorphous Li_xSi using machine-learning-assisted sampling with an evolutionary algorithm," J. Chem. Phys. **148**, 241711 (2018).

[20] J. Behler, "Neural network potential-energy surfaces in chemistry: A tool for large-scale simulations," Phys. Chem. Chem. Phys. **13**, 17930–17955 (2011).

[21] J. Behler and M. Parrinello, "Generalized neural-network representation of high-dimensional potential-energy surfaces," Phys. Rev. Lett. **98**, 146401 (2007).

[22] J. Behler, R. Martoňák, D. Donadio, and M. Parrinello, "Metadynamics simulations of the high-pressure phases of silicon employing a high-dimensional neural network potential," Phys. Rev. Lett. **100**, 185501 (2008).

[23] J. Behler, R. Martoňák, D. Donadio, and M. Parrinello, "Pressure-induced phase transitions in silicon studied by neural network-based metadynamics simulations," Phys. Status Solidi B **245**, 2618–2629 (2008).

[24] J. Behler, K. Reuter, and M. Scheffler, "Nonadiabatic effects in the dissociation of oxygen molecules at the Al(111) surface," Phys. Rev. B **77**, 115421 (2008).

[25] V. Botu, R. Batra, J. Chapman, and R. Ramprasad, "Machine learning force fields: Construction, validation, and outlook," J. Phys. Chem. C **121**, 511–522 (2016).

[26] V. Botu and R. Ramprasad, "Learning scheme to predict atomic forces and accelerate materials simulations," Phys. Rev. B **92**, 094306 (2015).

[27] V. Botu and R. Ramprasad, "Adaptive machine learning framework to accelerate *ab initio* molecular dynamics," Int. J. Quantum Chem. **115**, 1074–1083 (2015).

[28] M. A. Caro, A. Aarva, V. L. Deringer, G. Csányi, and T. Laurila, "Reactivity of amorphous carbon surfaces: Rationalizing the role of structural motifs in functionalization using machine learning," Chem. Mater. **30**, 7446–7455 (2018).

[29] W.-K. Chen, X.-Y. Liu, W. Fang, P. O. Dral, and G. Cui, "Deep learning for nonadiabatic excited-state dynamics," J. Phys. Chem. Lett. **9**, 6702–6708 (2018).

[30] S. Chmiela, H. E. Sauceda, K.-R. Müller, and A. Tkatchenko, "Towards exact molecular dynamics simulations with machine-learned force fields," Nat. Commun. **9**, 3887 (2018).

[31] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller, "Machine learning of accurate energy-conserving molecular force fields," Sci. Adv. **3**, e1603015 (2017).

[32] O. T. Unke, D. Koner, S. Patra, S. Käser, and M. Meuwly, "High-dimensional potential energy surfaces for molecular simulations: from empiricism to machine learning," Mach. Learn.: Sci. Technol. **1**, 013001 (2020).

[33] H. Eshet, R. Z. Khaliullin, T. D. Kuhne, J. Behler, and M. Parrinello, "*Ab initio* quality neural-network potential for sodium," Phys. Rev. B **81**, 184107 (2010).

[34] R. Galvelis and Y. Sugita, "Neural network and nearest neighbor algorithms for enhancing sampling of molecular dynamics," J. Chem. Theory Comput. **13**, 2489–2500 (2017).

[35] M. Gastegger, J. Behler, and P. Marquetand, "Machine learning molecular dynamics for the simulation of infrared spectra," Chem. Sci. **8**, 6924–6935 (2017).

[36] A. Glielmo, P. Sollich, and A. De Vita, "Accurate interatomic force fields via machine learning with covariant kernels," Phys. Rev. B **95**, 214302 (2017).

[37] C. M. Handley and P. L. A. Popelier, "Potential energy surfaces fitted by artificial neural networks," J. Phys. Chem. A **114**, 3371–3383 (2010).

[38] S. Houlding, S. Y. Liem, and P. L. A. Popelier, "A polarizable high-rank quantum topological electrostatic potential developed using neural networks: Molecular dynamics simulations on the hydrogen fluoride dimer," Int. J. Quantum Chem. **107**, 2817–2827 (2007).

[39] D. Hu, Y. Xie, X. Li, L. Li, and Z. Lan, "Inclusion of machine learning kernel ridge regression potential energy surfaces in on-the-fly nonadiabatic molecular dynamics simulation," J. Phys. Chem. Lett. **9**, 2725–2732 (2018).

[40] T. D. Huan, R. Batra, J. Chapman, S. Krishnan, L. Chen, and R. Ramprasad, "A universal strategy for the creation of machine learning-based atomistic force fields," npj Comput. Mater. **3**, 37 (2017).

[41] J.-X. Huang, G. Csányi, J.-B. Zhao, J. Cheng, and V. L. Deringer, "First-principles study of alkali-metal intercalation in disordered carbon anode materials," J. Mater. Chem. A **7**, 19070–19080 (2019).

[42] S. Jindal and S. S. Bulusu, "A transferable artificial neural network model for atomic forces in nanoparticles," J. Chem. Phys. **149**, 194101 (2018).

[43] R. Jinnouchi, J. Lahnsteiner, F. Karsai, G. Kresse, and M. Bokdam, "Phase transitions of hybrid perovskites simulated by machine-learning force fields trained on the fly with Bayesian inference," Phys. Rev. Lett. **122**, 225701 (2019).

[44] R. Jinnouchi, F. Karsai, and G. Kresse, "On-the-fly machine learning force field generation: Application to melting points," Phys. Rev. B **100**, 014105 (2019).

[45] A. Kamath, R. A. Vargas-Hernández, R. V. Krems, T. Carrington, Jr., and S. Manzhos, "Neural networks vs Gaussian process regression for representing potential energy surfaces: A comparative study of fit quality and vibrational spectrum accuracy," J. Chem. Phys. **148**, 241702 (2018).

[46] R. Z. Khaliullin, H. Eshet, T. D. Kühne, J. Behler, and M. Parrinello, "Nucleation mechanism for the direct graphite-to-diamond phase transition," Nat. Mater. **10**, 693–697 (2011).

[47] B. Kolb, B. Zhao, J. Li, B. Jiang, and H. Guo, "Permutation invariant potential energy surfaces for polyatomic reactions using atomistic neural networks," J. Chem. Phys. **144**, 224103 (2016).

[48] R. V. Krems, "Bayesian machine learning for quantum molecular dynamics," Phys. Chem. Chem. Phys. **21**, 13392–13410 (2019).

[49] H. M. Le, S. Huynh, and L. M. Raff, "Molecular dissociation of hydrogen peroxide (HOOH) on a neural network *ab initio* potential surface with a new configuration sampling method involving gradient fitting," J. Chem. Phys. **131**, 014107 (2009).

[50] H. M. Le and L. M. Raff, "*Cis $\rightarrow$ trans, trans $\rightarrow$ cis* isomerizations and N–O bond dissociation of nitrous acid (HONO) on an *ab initio* potential surface obtained by novelty sampling and feed-forward neural network fitting," J. Chem. Phys. **128**, 194310 (2008).

[51] H. M. Le and L. M. Raff, "Molecular dynamics investigation of the bimolecular reaction $BeH + H_2 \rightarrow BeH_2 + H$ on an *ab initio* potential-energy surface obtained using neural network methods with both potential and gradient accuracy determination," J. Phys. Chem. A **114**, 45–53 (2010).

[52] P. Li, X. Jia, X. Pan, Y. Shao, and Y. Mei, "Accelerated computation of free energy profile at *ab initio* quantum mechanical/molecular mechanics accuracy via a semi-empirical reference potential. I. Weighted thermodynamics perturbation," J. Chem. Theory Comput. **14**, 5583–5596 (2018).

[53] Z. Li, J. R. Kermode, and A. De Vita, "Molecular dynamics with on-the-fly machine learning of quantum-mechanical forces," Phys. Rev. Lett. **114**, 096405 (2015).

[54] Q. Liu, D. Lu, and M. Chen, "Structure and dynamics of warm dense aluminum: A molecular dynamics study with density functional theory and deep potential," J. Phys.: Condens. Matter. **32**, 144002 (2020).

[55] S. Lorenz, A. Groß, and M. Scheffler, "Representing high-dimensional potential-energy surfaces for reactions at surfaces by neural networks," Chem. Phys. Lett. **395**, 210–215 (2004).

[56] S. Lorenz, M. Scheffler, and A. Gross, "Descriptions of surface chemical reactions using a neural network representation of the potential-energy surface," Phys. Rev. B **73**, 115431 (2006).

[57] J. Ludwig and D. G. Vlachos, "Ab initio molecular dynamics of hydrogen dissociation on metal surfaces using neural networks and novelty sampling," J. Chem. Phys. **127**, 154716 (2007).

[58] M. Malshe, L. M. Raff, M. G. Rockley, M. Hagan, P. M. Agrawal, and R. Komanduri, "Theoretical investigation of the dissociation dynamics of vibrationally excited vinyl bromide on an ab initio potential-energy surface obtained using modified novelty sampling and feedforward neural networks. II. Numerical application of the method," J. Chem. Phys. **127**, 134105 (2007).

[59] S. Manzhos, R. Dawes, and T. Carrington, Jr., "Neural network-based approaches for building high dimensional and quantum dynamics-friendly potential energy surfaces," Int. J. Quantum Chem. **115**, 1012–1020 (2015).

[60] T. Morawietz, V. Sharma, and J. Behler, "A neural network potential-energy surface for the water dimer based on environment-dependent atomic energies and charges," J. Chem. Phys. **136**, 064103 (2012).

[61] E. V. Podryabinkin and A. V. Shapeev, "Active learning of linearly parametrized interatomic potentials," Comput. Mater. Sci. **140**, 171–180 (2017).

[62] A. Pukrittayakamee, M. Malshe, M. Hagan, L. M. Raff, R. Narulkar, S. Bukkapatnum, and R. Komanduri, "Simultaneous fitting of a potential-energy surface and its corresponding force fields using feedforward neural networks," J. Chem. Phys. **130**, 134101 (2009).

[63] V. Quaranta, M. Hellström, J. Behler, J. Kullgren, P. D. Mitev, and K. Hermansson, "Maximally resolved anharmonic OH vibrational spectrum of the water/ZnO(1010) interface from a high-dimensional neural network potential," J. Chem. Phys. **148**, 241720 (2018).

[64] L. M. Raff, M. Malshe, M. Hagan, D. I. Doughan, M. G. Rockley, and R. Komanduri, "Ab initio potential-energy surfaces for complex, multichannel systems using modified novelty sampling and feedforward neural networks," J. Chem. Phys. **122**, 084104 (2005).

[65] G. W. Richings and S. Habershon, "Direct quantum dynamics using grid-based wave function propagation and machine-learned potential energy surfaces," J. Chem. Theory Comput. **13**, 4012–4024 (2017).

[66] K. T. Schütt, P. Kessel, M. Gastegger, K. A. Nicoli, A. Tkatchenko, and K.-R. Müller, "SchNetPack: A deep learning toolbox for atomistic systems," J. Chem. Theory Comput. **15**, 448–455 (2019).

[67] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, "SchNet—A deep learning architecture for molecules and materials," J. Chem. Phys. **148**, 241722 (2018).

[68] K. Shao, J. Chen, Z. Zhao, and D. H. Zhang, "Communication: Fitting potential energy surfaces with fundamental invariant neural network," J. Chem. Phys. **145**, 071101 (2016).

[69] A. S. Christensen, L. A. Bratholm, F. A. Faber, and O. A. von Lilienfeld, "FCHL revisited: Faster and more accurate quantum machine learning," J. Chem. Phys. **152**, 044107 (2020).

[70] G. C. Sosso and M. Bernasconi, "Harnessing machine learning potentials to understand the functional properties of phase-change materials," MRS Bull. **44**, 705–709 (2019).

[71] T. Suzuki, R. Tamura, and T. Miyazaki, "Machine learning for atomic forces in a crystalline solid: Transferability to various temperatures," Int. J. Quantum Chem. **117**, 33–39 (2017).

[72] O. T. Unke and M. Meuwly, "Physnet: A neural network for predicting energies, forces, dipole moments, and partial charges," J. Chem. Theory Comput. **15**, 3678–3693 (2019).

[73] R. A. Vargas-Hernández, Y. Guan, D. H. Zhang, and R. V. Krems, "Bayesian optimization for the inverse scattering problem in quantum reaction dynamics," New J. Phys. **21**, 022001 (2019).

[74] H. Wang and W. Yang, "Force field for water based on neural network," J. Phys. Chem. Lett. **9**, 3232–3240 (2018).

[75] H. Wang, L. Zhang, J. Han, and W. E, "A deep learning package for many-body potential energy representation and molecular dynamics," Comput. Phys. Commun. **228**, 178–184 (2018).

[76] J. Westermayr, M. Gastegger, M. F. S. J. Menger, S. Mai, L. González, and P. Marquetand, "Machine learning enables long time scale molecular photodynamics simulations," Chem. Sci. **10**, 8100–8107 (2019).

[77] K. Yao, J. E. Herr, D. W. Toth, R. McKintyre, and J. Parkhill, "The TensorMol-0.1 model chemistry: A neural network augmented with long-range physics," Chem. Sci. **9**, 2261–2269 (2018).

[78] L. Zhang, J. Han, H. Wang, R. Car, and W. E, "Deep potential molecular dynamics: A scalable model with the accuracy of quantum mechanics," Phys. Rev. Lett. **120**, 143001 (2018).

[79] Y. Zhang, X. Zhou, and B. Jiang, "Bridging the gap between direct dynamics and globally accurate reactive potential energy surfaces using neural networks," J. Phys. Chem. Lett. **10**, 1185–1191 (2019).

[80] G. Schmitz, D. G. Artiukhin, and O. Christiansen, "Approximate high mode coupling potentials using Gaussian process regression and adaptive density guided sampling," J. Chem. Phys. **150**, 131102 (2019).

[81] A. P. Bartók, M. J. Gillan, F. R. Manby, and G. Csányi, "Machine-learning approach for one- and two-body corrections to density functional theory: Applications to molecular and condensed water," Phys. Rev. B **88**, 054104 (2013).

[82] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld, "Big Data meets quantum chemistry approximations: The Δ-machine learning approach," J. Chem. Theory Comput. **11**, 2087–2096 (2015).

[83] P. Zaspel, B. Huang, H. Harbrecht, and O. A. von Lilienfeld, "Boosting quantum machine learning models with a multilevel combination technique: Pople diagrams revisited," J. Chem. Theory Comput. **15**, 1546–1559 (2018).

[84] G. Sun and P. Sautet, "Toward fast and reliable potential energy surfaces for metallic Pt clusters by hierarchical delta neural networks," J. Chem. Theory Comput. **15**, 5614–5627 (2019).

[85] R. Ramakrishnan, M. Hartmann, E. Tapavicza, and O. A. von Lilienfeld, "Electronic spectra from TDDFT and machine learning in chemical space," J. Chem. Phys. **143**, 084111 (2015).

[86] L. Shen, J. Wu, and W. Yang, "Multiscale quantum mechanics/molecular mechanics simulations with neural networks," J. Chem. Theory Comput. **12**, 4934–4946 (2016).

[87] L. Shen and W. Yang, "Molecular dynamics simulations with quantum mechanics/molecular mechanics and adaptive neural networks," J. Chem. Theory Comput. **14**, 1442–1455 (2018).

[88] L. Ward, B. Blaiszik, I. Foster, R. S. Assary, B. Narayanan, and L. Curtiss, "Machine learning prediction of accurate atomization energies of organic molecules from low-fidelity quantum chemical calculations," MRS Commun. **9**, 891–899 (2019).

[89] M. Welborn, L. Cheng, and T. F. Miller, "Transferability in machine learning for electronic structure via the molecular orbital basis," J. Chem. Theory Comput. **14**, 4772–4779 (2018).

[90] L. Cheng, M. Welborn, A. S. Christensen, and T. F. Miller, "A universal density matrix functional from molecular orbital-based machine learning: Transferability across organic molecules," J. Chem. Phys. **150**, 131103 (2019).

[91] R. M. Balabin and E. I. Lomakina, "Neural network approach to quantum-chemistry data: Accurate prediction of density functional theory energies," J. Chem. Phys. **131**, 074104 (2009).

[92] R. M. Balabin and E. I. Lomakina, "Support vector machine regression (LS-SVM)—An alternative to artificial neural networks (ANNs) for the analysis of quantum chemistry data?," Phys. Chem. Chem. Phys. **13**, 11710–11718 (2011).

[93] X.-M. Duan, Z.-H. Li, G.-L. Song, W.-N. Wang, G.-H. Chen, and K.-N. Fan, "Neural network correction for heats of formation with a larger experimental training set and new descriptors," Chem. Phys. Lett. **410**, 125–130 (2005).

[94] L. H. Hu, X. J. Wang, L. H. Wong, and G. H. Chen, "Combined first-principles calculation and neural-network correction approach for heat of formation," J. Chem. Phys. **119**, 11501–11507 (2003).

[95]H. Li, L. Shi, M. Zhang, Z. Su, X. Wang, L. Hu, and G. Chen, "Improving the accuracy of density-functional theory calculation: The genetic algorithm and neural network approach," J. Chem. Phys. **126**, 144101 (2007).

[96]M. Malshe, A. Pukrittayakamee, L. M. Raff, M. Hagan, S. Bukkapatnam, and R. Komanduri, "Accurate prediction of higher-level electronic structure energies for large databases using neural networks, Hartree–Fock energies, and small subsets of the database," J. Chem. Phys. **131**, 124127 (2009).

[97]J. T. Margraf and K. Reuter, "Making the coupled cluster correlation energy machine-learnable," J. Phys. Chem. A **122**, 6343–6348 (2018).

[98]G. M. e Silva, P. H. Acioli, and A. C. Pedroza, "Estimating correlation energy of diatomic molecules and atoms with neural networks," J. Comput. Chem. **18**, 1407–1414 (1997).

[99]J. Sun, J. Wu, T. Song, L. Hu, K. Shan, and G. Chen, "Alternative approach to chemical accuracy: A neural networks-based first-principles method for heat of formation of molecules made of H, C, N, O, F, S, and Cl," J. Phys. Chem. A **118**, 9120–9131 (2014).

[100]X. Wang, L. Hu, L. Wong, and G. Chen, "A combined first-principles calculation and neural networks correction approach for evaluating Gibbs energy of formation," Mol. Simul. **30**, 9–15 (2004).

[101]J. Wu and X. Xu, "The X1 method for accurate and efficient prediction of heats of formation," J. Chem. Phys. **127**, 214105 (2007).

[102]J. Wu and X. Xu, "Improving the B3LYP bond energies by using the X1 method," J. Chem. Phys. **129**, 164103 (2008).

[103]A. Owens, S. N. Yurchenko, A. Yachmenev, J. Tennyson, and W. Thiel, "Accurate *ab initio* vibrational energies of methyl chloride," J. Chem. Phys. **142**, 244306 (2015).

[104]C. Tsallis, "Possible generalization of Boltzmann–Gibbs statistics," J. Stat. Phys. **52**, 479–487 (1998).

[105]C. Tsallis and D. A. Stariolo, "Generalized simulated annealing," Physica A **233**, 395–406 (1996).

[106]Y. Xiang, D. Y. Sun, W. Fan, and X. G. Gong, "Generalized simulated annealing algorithm and its application to the Thomson model," Phys. Lett. A **233**, 216–220 (1997).

[107]Y. Xiang and X. G. Gong, "Generalized simulated annealing algorithm and its application to the Thomson model," Phys. Rev. E **62**, 4473 (2000).

[108]Y. Xiang, S. Gubian, B. Suomela, J. Hoeng, and X. G. Gong, "Generalized simulated annealing for efficient global optimization: The GenSA package," R J. **5**(1), 13–28 (2013).

[109]K. Mullen, "Continuous Global Optimization in R," J. Stat. Software **60**, 1–45 (2014).

[110]P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and Contributors of SciPy

1.0, "SciPy 1.0: Fundamental algorithms for scientific computing in Python," Nat. Methods **17**, 261–272 (2020).

[111]E. Jones, T. Oliphant, P. Peterson *et al.*, SciPy: Open Source Scientific Tools for Python, 2001, http://www.scipy.org/; accessed 20 February 2020.

[112]J. A. Nelder and R. Mead, "A simplex method for function minimization," Comput. J. **7**, 308–313 (1965).

[113]M. H. Wright, *Direct Search Methods: Once Scorned, Now Respectable*, Pitman Research Notes in Mathematics Series (Addison Wesley Longman, Harlow, UK, 1996), pp. 191–208.

[114]J. Nocedal and S. Wright, *Numerical Optimization*, 2nd ed., Pitman Research Notes in Mathematics Series (Springer Science & Business Media, 2006).

[115]T. B. Adler, G. Knizia, and H.-J. Werner, "A simple and efficient CCSD(T)-F12 approximation," J. Chem. Phys. **127**, 221106 (2007).

[116]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).

[117]A. Owens, A. Yachmenev, W. Thiel, A. Fateev, J. Tennyson, and S. N. Yurchenko, "ExoMol line lists—XXIX. The rotation-vibration spectrum of methyl chloride up to 1200 K," Mon. Not. R. Astron. Soc. **479**, 3002–3010 (2018).

[118]R. A. Kendall, T. H. Dunning, Jr., and R. J. Harrison, "Electron affinities of the first-row atoms revisited. Systematic basis sets and wave functions," J. Chem. Phys. **96**, 6796–6806 (1992).

[119]D. E. Woon and T. H. Dunning, Jr., "Gaussian basis sets for use in correlated molecular calculations. III. The atoms aluminum through argon," J. Chem. Phys. **98**, 1358–1371 (1993).

[120]H.-J. Werner, P. J. Knowles, G. Knizia, F. R. Manby, and M. Schütz, "Molpro: A general-purpose quantum chemistry program package," Wiley Interdiscip. Rev.: Comput. Mol. Sci. **2**, 242–253 (2012).

[121]H. Partridge and D. W. Schwenke, "The determination of an accurate isotope dependent potential energy surface for water from extensive *ab initio* calculations and experimental data," J. Chem. Phys. **106**, 4618–4639 (1997).

[122]P. O. Dral, "MLatom: A program package for quantum chemical research assisted by machine learning," J. Comput. Chem. **40**, 2339–2347 (2019).

[123]A. P. Bartók, S. De, C. Poelking, N. Bernstein, J. R. Kermode, G. Csányi, and M. Ceriotti, "Machine learning unifies the modeling of materials and molecules," Sci. Adv. **3**, e1701816 (2017).

[124]P. O. Dral, M. Barbatti, and W. Thiel, "Nonadiabatic excited-state dynamics with machine learning," J. Phys. Chem. Lett. **9**, 5660–5663 (2018).

[125]P. O. Dral, MLatom: A Package for Atomistic Simulations with Machine Learning, Development Version, Xiamen University, Xiamen, China, 2013–2020, http://MLatom.com; accessed 23 April 2020.

[126]P. O. Dral, A. Owens, A. Dral, and G. Csányi, "Data set with *ab initio* CH_3Cl potential energy surface," figshare at https://doi.org/10.6084/m9.figshare.11876364.