Gradient domain machine learning with composite kernels: improving the accuracy of PES and force fields for large molecules

MACHINE
LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
10 September 2021

REVISED
5 November 2021

ACCEPTED FOR PUBLICATION
10 November 2021

PUBLISHED
25 November 2021

PAPER

Gradient domain machine learning with composite kernels: improving the accuracy of PES and force fields for large molecules

K Asnaashari* and R V Krems

Department of Chemistry, University of British Columbia, Vancouver, BC V6T 1Z1, Canada

* Author to whom any correspondence should be addressed.

E-mail: kasnaashari@chem.ubc.ca

Keywords: GDML, Gaussian process regression, potential energy surface, force field, machine learning, kernel methods

Abstract

The generalization accuracy of machine learning models of potential energy surfaces (PES) and force fields (FF) for large polyatomic molecules can be improved either by increasing the number of training points or by improving the models. In order to build accurate models based on expensive *ab initio* calculations, much of recent work has focused on the latter. In particular, it has been shown that gradient domain machine learning (GDML) models produce accurate results for high-dimensional molecular systems with a small number of *ab initio* calculations. The present work extends GDML to models with composite kernels built to maximize inference from a small number of molecular geometries. We illustrate that GDML models can be improved by increasing the complexity of underlying kernels through a greedy search algorithm using Bayesian information criterion as the model selection metric. We show that this requires including anisotropy into kernel functions and produces models with significantly smaller generalization errors. The results are presented for ethanol, uracil, malonaldehyde and aspirin. For aspirin, the model with composite kernels trained by forces at 1000 randomly sampled molecular geometries produces a global 57-dimensional PES with the mean absolute accuracy 0.177 kcal mol^{-1} (61.9 cm^{-1}) and FFs with the mean absolute error 0.457 kcal mol^{-1} Å^{-1}.

1. Introduction

Accurate potential energy surfaces (PES) and force fields (FF) are required for simulations of dynamics of molecules. A major recent effort has been to develop accurate models of PES and FFs for large polyatomic molecules with accuracy of *ab initio* calculations. As the complexity of molecules grows, it becomes increasingly difficult to produce accurate analytical fits of PES and FFs as choosing suitable functions for the parameterization becomes challenging. This problem can be addressed with machine learning (ML), as illustrated by a large body of recent work on neural network [1–14] and kernel regression [14–34] models of PES and FFs. These ML models are generally trained by potential energies and/or forces computed with *ab initio* methods for different molecular geometries. The accuracy of ML models generally increases with the number of training data. However, *ab initio* calculations are expensive. Therefore, a significant focus of recent work has been on building accurate ML models with as few *ab initio* calculations as possible [35–41].

For problems with a small number of training points, kernel regression models have been shown to produce accurate PES and outperform NNs in some cases [21]. The accuracy of kernel models largely depends on (a) the descriptors used for the input variables [42, 43]; (b) the type of model; and (c) the mathematical form of the kernel [28, 29, 44–49]. Of different model approaches, gradient domain machine learning (GDML) has so far proven to yield the most accurate results for molecules with ~10–57 degrees of freedom when the number of training molecular geometries is restricted to $\lesssim$5000 [38–41]. GDML models are trained by forces or by combinations of forces and energies to produce accurate FFs and PES. The accuracy of GDML models can be improved by building molecular symmetries into underlying kernels, which produces symmetrized models [40], hereafter denoted as sGDML. Symmetrization effectively reduces the size of the input space. However, all of the previous GDML calculations have been performed with simple, isotropic kernels that do not discriminate between input dimensions.

The effect of kernel complexity on the accuracy of PES has been explored in applications of Gaussian process (GP) regression to both low (4 atoms [28]) and high (19 atoms [29]) dimensional molecules. Sugisawa *et al* [28] and Dai and Krems [29] showed that the accuracy of the GP models of PES can be enhanced by increasing the complexity of underlying kernels through a greedy algorithm combining simple mathematical functions guided by the Bayesian information criterion (BIC) as the model selection metric. This kernel selection algorithm is based on an earlier work demonstrating GP models with enhanced prediction power for pattern recognition problems [44, 45] and applications of GP models with composite kernels to extrapolation of properties of quantum systems in Hamiltonian parameter spaces [46]. In the present work we combine the kernel construction method of [28, 29, 44–46] with the GDML approach to improve the accuracy of GDML models. We demonstrate that the resulting models benefit simultaneously from the GDML formalism based on training models with forces and from the BIC-guided model selection approach.

Previous GDML models were built as kernel ridge regression models with the kernel parameters determined by cross-validation using grid search [38–40]. The grid-search approach is suitable for simple, isotropic kernels that depend on a small number of parameters. In order to take advantage of BIC, one needs to ensure that models can be trained by maximizing log marginal likelihood (LML). The first result of the present work illustrates that it is necessary to include kernel anisotropy in order to train GDML models by LML maximization. Our analysis shows that LML, and hence BIC, is not a good metric for model selection in GDML with isotropic kernels. Once the kernel anisotropy is included, however, the BIC can be used to enhance the complexity of the GDML kernels. We build composite kernels for four different molecules and illustrate the effect of kernel complexity on improving the accuracy of GDML predictions. In some instances the resulting GDML models are shown to be more accurate than the sGDML predictions.

The remainder of this paper is organized as follows. The subsequent section begins with a brief summary of the notation used throughout this article and the description of the ML methods. Section 3 presents the results by first discussing the effect of the kernel anisotropy and then the kernel complexity. The results are presented for the PES and FFs for four molecules: ethanol (9 atoms, 21 degrees of freedom), malonaldehyde (9 atoms, 21 degrees of freedom), uracil (12 atoms, 30 degrees of freedom), and aspirin (21 atoms, 57 degrees of freedom). The accuracy of the corresponding models is compared with the previous GDML and sGDML calculations. We show that the GDML model with composite kernels produces a global 57-dimensional PES for aspirin with the mean absolute accuracy 0.177 kcal mol^{-1} (61.9 cm^{-1}) and FF with the mean absolute accuracy 0.457 kcal mol^{-1} Å^{-1} when trained by 1000 randomly sampled molecular geometries.

2. Methods

2.1. Model abbreviations

We present and discuss results for several ML models of PES and FFs. Table 1 lists the acronyms used throughout this work. The accuracy of the models is quantified by the mean absolute error (MAE):

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{f}(\boldsymbol{x}_i)| \tag{1}$$

and the root mean squared error (RMSE):

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}(\boldsymbol{x}_i)\right)^2} \tag{2}$$

evaluated on a hold-out set of n potential energies or FFs, randomly sampled from the entire configuration space and not used for training the models. One exception is the training energy error, hereafter denoted 'Train E', that is computed as the RMSE with the energy points in the training set. In equation (1), $\boldsymbol{x}_i$ is a p-dimensional vector specifying the positions of atoms in a molecule, y_i represents the potential energy or forces experienced by each individual atom at $\boldsymbol{x}_i$, and $\hat{f}(\boldsymbol{x}_i)$ denotes the prediction of the ML model at $\boldsymbol{x}_i$. We build models $\hat{f}_{\mathrm{E}}$ for energy and $\hat{f}_{\mathrm{F}}$ for forces.

2.2. GDML

GDML models explicitly construct an energy-conserving FF by implementing the relation between the energy of the molecule and the forces acting on each atom as an *a priori* condition of the model [38]:

$$\hat{f}_{\mathrm{F}(\boldsymbol{x})} = -\nabla \hat{f}_{\mathrm{E}(\boldsymbol{x})}, \tag{3}$$

Table 1. Abbreviations used in this work.

Abbreviation	Meaning
GDML	Gradient domain machine learning
sGDML	GDML with symmetrized kernels
AGDML	GDML with simple anisotropic kernels
AGDML(c)	GDML with composite anisotropic kernels
MAE	Mean absolute error
RMSE	Root mean squared error
LML	Log marginal likelihood
Train E	Training energy error (RMSE)

where for a molecular system of N atoms, $\boldsymbol{x} \in \chi^{3N}$ represents the coordinates of the atoms, $\hat{f}_{\mathrm{E}}(\boldsymbol{x}) : \chi^{3N} \to \mathbb{R}$ is an estimator of the energy, $\hat{f}_{\mathrm{F}(\boldsymbol{x})} : \chi^{3N} \to \mathbb{R}^{3N}$ is an estimator of the forces, and ∇ is the gradient operator. If we consider the energy estimator as a realization of a GP (4), since ∇ is a linear operator, the estimator of forces will also be a realization of a GP (5):

$$\hat{f}_{\mathrm{E}} \sim \mathrm{GP}\left[\mu(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x'})\right] \tag{4}$$

$$\hat{f}_{\mathrm{F}} \sim \mathrm{GP}\left[-\nabla\mu(\boldsymbol{x}), \nabla_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{x'}) \nabla^{T}_{\boldsymbol{x'}}\right] \tag{5}$$

where $\mu(\boldsymbol{x}) : \chi^{3N} \to \mathbb{R}$ and $k(\boldsymbol{x}, \boldsymbol{x'}) : \chi^{3N} \times \chi^{3N} \to \mathbb{R}$ are the mean and covariance functions of the GP. One can also model both forces and energies as a single GP through the methodology described by Solak *et al* [50]:

$$\hat{f}_{\mathrm{FE}} \sim \mathrm{GP}\left[\begin{bmatrix}\nabla\mu(\boldsymbol{x}) \\ \mu(\boldsymbol{x})\end{bmatrix}, \begin{bmatrix}\nabla_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{x'})\nabla^{T}_{\boldsymbol{x'}} & \nabla_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{x'}) \\ k(\boldsymbol{x}, \boldsymbol{x'})\nabla^{T}_{\boldsymbol{x'}} & k(\boldsymbol{x}, \boldsymbol{x'})\end{bmatrix}\right]. \tag{6}$$

The models given by equation (6) require both forces and energies for each molecular configuration in the training data, while the models in equation (5) require only the mean of the energies in the training set. Previous studies have shown that these hybrid models overfit the energies at the cost of the FF accuracy [41]. We observed that both types of models when trained using LML yield very similar predictions for the same training sets. Therefore, in what follows, we use models trained by forces only, i.e. models given by equation (5).

Equation (5) describes a multi-output GP which predicts the vector components of forces for each atom in a molecule. The covariance function of the derivative of a GP is the second derivative of the original kernel function and $\nabla_{\boldsymbol{x}} k \nabla^{T}_{\boldsymbol{x'}} = \mathrm{Hess}_{\boldsymbol{x}}(k) = k_{\mathrm{H}}(\boldsymbol{x}, \boldsymbol{x'}) \in \mathbb{R}^{3N\times 3N}$ for stationary kernels. The posterior mean of the model of forces is

$$\hat{f}_{\mathrm{F}}(\mathbf{X}^{*}) = \alpha k_{\mathrm{H}}(\mathbf{X}^{*}, \mathbf{X})^{T} = \sum_{i}^{M}\sum_{j}^{3N}(\alpha_i)_j \frac{\partial}{\partial x_j}\nabla k(\mathbf{X}^{*}, \boldsymbol{x}_i) \tag{7}$$

where $\mathbf{X} \in \mathbb{R}^{M\times 3N}$ are the training geometries (three coordinates for each of N atoms of M training geometries), $\mathbf{X}^{*} \in \mathbb{R}^{M'\times 3N}$ are the molecular geometries corresponding to $M^{'}$ evaluation points in the configuration space, $k_{\mathrm{H}}(\mathbf{X}^{*}, \mathbf{X}) \in \mathbb{R}^{3M'N\times 3MN}$ is the kernel matrix coupling the training and evaluation geometries, $\frac{\partial}{\partial x_j}$ is a partial derivative with respect to the jth dimension of $\boldsymbol{x}_i$ and $\boldsymbol{\alpha} \in \mathbb{R}^{3MN}$ is defined as:

$$\boldsymbol{\alpha} \equiv (K_{\mathrm{H}} + \lambda\mathbb{I})^{-1}\boldsymbol{y}_{\mathrm{F}} \tag{8}$$

where $K_{\mathrm{H}} = k_{\mathrm{H}}(\mathbf{X}, \mathbf{X}) \in \mathbb{R}^{3NM\times 3NM}$, $\boldsymbol{y}_{\mathrm{F}} \in \mathbb{R}^{3NM}$ are the components of the forces in the training set flattened into a one-dimensional vector and λ is the parameter related to the variance of the Gaussian noise of the training data (equivalent to the regularization parameter in kernel ridge regression). While the training data in this work are noiseless, the parameter λ will be shown to play an important role due to overcompletness of the descriptors (see more details in section 3.1).

By integrating equation (7), one obtains the model of the PES:

$$\hat{f}_{\mathrm{E}}(\mathbf{X}^{*}) = \boldsymbol{\alpha} k_{\mathrm{G}}(\mathbf{X}^{*}, \mathbf{X})^{T} = \sum_{i}^{M}\sum_{j}^{3N}(\alpha_i)_j \frac{\partial}{\partial x_j} k(\mathbf{X}^{*}, \boldsymbol{x}_i) + c \tag{9}$$

where k_{G} is defined as the gradient of the kernel function with respect to the input dimensions $k_{\mathrm{G}}(\mathbf{X}^*, \mathbf{X}) \in \mathbb{R}^{M' \times 3MN}$ and c is the integration constant which can be calculated as

$$c = \frac{1}{M} \sum_{i}^{M} \left[E_i + \hat{f}_{\mathrm{E}}(\boldsymbol{x}_i) \right], \tag{10}$$

where E_i is the energy of the ith geometry of the training set.

Given the kernel matrix and a value of λ, the logarithm of marginal likelihood (LML) can be calculated for these models as follows:

$$\log p(\boldsymbol{y}_{\mathrm{F}} | \mathbf{X}) = -\frac{1}{2} \boldsymbol{y}_f{}^T (K_{\mathrm{H}} + \lambda \mathbb{I})^{-1} \boldsymbol{y}_{\mathrm{F}} - \frac{1}{2} \log |K_{\mathrm{H}} + \lambda \mathbb{I}| - \frac{M}{2} \log 2\pi. \tag{11}$$

Note again that in this formulation the model does not use energy points directly so energies are not used for building the PES. Energy predictions are determined by the individual energy points indirectly, through forces, and through the mean of the energies (first term in equation (10)). LML for such models is consequently independent of the energies and is written in terms of forces $\boldsymbol{y}_{\mathrm{F}}$ only.

2.3. Kernel functions in GDML

The GDML method described above can, in principle, be used with any doubly differentiable stationary kernel function. Some examples of such kernel functions commonly used for kernel regression models include Matérn functions of order $\geqslant 5/2$, radial basis functions (RBF) or rational quadratic (RQ) functions [45]:

$$\text{RBF:} \quad k(\boldsymbol{x}, \boldsymbol{x'}) = \sigma \exp\left(-\frac{1}{2} r^2(\boldsymbol{x}, \boldsymbol{x'}) \right) \tag{12}$$

$$\text{Matérn 5/2:} \quad k(\boldsymbol{x}, \boldsymbol{x'}) = \sigma \left(1 + \sqrt{5} r(\boldsymbol{x}, \boldsymbol{x'}) + \frac{5}{3} r^2(\boldsymbol{x}, \boldsymbol{x'}) \right) \exp\left(-\sqrt{5} r(\boldsymbol{x}, \boldsymbol{x'}) \right) \tag{13}$$

$$\text{RQ:} \quad k(\boldsymbol{x}, \boldsymbol{x'}) = \sigma \left(1 + \frac{1}{2\alpha} r^2(\boldsymbol{x}, \boldsymbol{x'}) \right)^{-\alpha} \tag{14}$$

where $r^2(\boldsymbol{x}, \boldsymbol{x'}) = (\boldsymbol{x} - \boldsymbol{x'})^T \times \Lambda \times (\boldsymbol{x} - \boldsymbol{x'})$ and Λ is a diagonal matrix. For isotropic kernels, $\Lambda = l \times \mathbb{I}$ with l being a positive scalar and $\mathbb{I}$ an identity matrix. To the best of our knowledge, previous studies and extensions of the GDML models [38–40] use the isotropic Matérn 5/2 kernel function. In the previous studies, σ in equation (13) was set to 1 and the single value of l was determined by grid search using cross-validation.

The main goal of the present work is to extend the previous GDML work to include composite kernel functions built using the methodology developed by Duvenaud *et al* [44–46]. We show below that this requires allowing for kernel anisotropy. For anisotropic kernels, Λ has a free parameter for each dimension of $\boldsymbol{x}$. Given the large number of descriptor dimensions for the molecules considered here (up to 57 degrees of freedom for aspirin, translating into 210 descriptor dimensions using the descriptors discussed below), allowing for kernel anisotropy makes grid search of kernel parameters impossible. This, however, does not present a problem when models are trained by LML maximization.

2.3.1. Simple kernels

We refer to kernels given by equations (12)–(14) as simple, whereas any linear combination of different simple kernels is hereafter referred to as a composite kernel function. For reasons described in the previous sections, we limit simple kernels to doubly differentiable stationary kernel functions and specifically to the set of three kernel functions given by equations (12)–(14). We consider models with simple kernels that are either isotropic or anisotropic. As specified in table 1, GDML models with simple anisotropic kernel functions will be denoted by AGDML. A model for aspirin based on a simple anisotropic kernel given by equation (13) has ≈210 trainable parameters. Note that the number of trainable parameters increases substantially as composite kernels are formed from simple kernels.

2.3.2. Composite kernels

As shown previously, composite kernel functions cannot be constructed as random combinations of simple kernels [51]. Instead, the kernel construction algorithm should be guided by a model selection metric. In this

Table 2. Number of configurations and the range of energies and forces for each of the four molecules in the MD17 dataset [1] used in this work.

Molecule	# of configurations	Minimum energy (kcal mol^{-1})	Maximum energy (kcal mol^{-1})	Minimum force (kcal mol^{-1} Å^{-1})	Maximum force (kcal mol^{-1} Å^{-1})
Ethanol	555 092	−97 208.4	−97 171.8	−211.1	220.9
Malonaldehyde	993 237	−167 514.2	−167 470.4	−286.0	284.6
Uracil	133 770	−260 120.6	−260 080.8	−237.3	239.2
Aspirin	211 762	−406 757.6	−406 702.3	−210.4	213.4

work, composite kernel functions are built using the methodology of [45] that involves a greedy search algorithm with the BIC as the model selection metric. The BIC is defined as follows:

$$\mathrm{BIC} = \max(\mathrm{LML}) - \frac{\gamma}{2}\log(M) \tag{15}$$

where γ is the number of free parameters in the model and M is the number of training geometries. This kernel selection algorithm can be viewed as a search tree with the following steps:

(a) Build models using simple kernel functions;
(b) Evaluate the BIC for each model;
(c) Select the model with the largest value of BIC as a base model;
(d) Add and multiply the kernel of the base model with each of the simple kernel functions used in step (a) to create new kernel functions and train the models with the new composite kernel functions;
(e) Repeat steps (b)–(d) until the improvements in test (validation) error become negligible or the error begins to rise due to overfitting.

Each iteration of steps (b)–(d) creates a new layer of the search tree. As the algorithm progresses to deeper layers, the complexity of the optimal kernel function increases.

This algorithm has been shown to increase the accuracy of PES of molecules using energy-based models [28, 29]. Here, we use this methodology with anisotropic kernel functions to build GDML models based on forces. Since we work with gradients and Hessians of the kernel functions, only linear combinations of kernel functions is considered in step (d) to simplify the kernel search in the present work. As specified in table 1, GDML models with anisotropic composite kernels are referred to AGDML(c).

3. Results

All results in this work use the MD17 dataset [1]. The molecules considered include ethanol, malonaldehyde, uracil, and aspirin. The energies and forces in the dataset were computed using the PBE + vdW-TS electronic structure method [52, 53]. Table 2 displays the number of configurations and the range of energies and forces in the datasets. We do not consider some of the molecules in MD17 dataset (benzene, naphthalene, and salicylic acid). These molecules are rigid and as a result their PES and FFs are quite simple to learn. Due to their simplicity, these molecules are well modeled by basic GDML (as evidenced by the fact that extension to sGDML does not result in significant accuracy improvements [40]) and are not expected to benefit significantly from AGDML.

3.1. Effects of kernel anisotropy

We begin by considering the effects of kernel anisotropy in models with simple kernels. The kernel anisotropy is included by increasing the number of trainable parameters of the kernel function to match the number of the input variables. Following previous work [38, 41], we use the following descriptor of molecules based on the Coulomb matrix:

$$D_{ij} = \begin{cases} \|\boldsymbol{r}_i - \boldsymbol{r}_j\|^{-1} & \text{for } i > j \\ 0 & \text{for } i \leqslant j \end{cases} \tag{16}$$

where $\boldsymbol{r}_i$ is the vector of coordinates of the ith atom in the molecule. Coulomb matrices benefit from the roto-translational invariance of the molecular systems. For a molecule with N atoms, there are $3\,N$ input dimensions (atom coordinates) and $N(N-1)/2$ descriptor dimensions. As each molecule has $3N-6$ independent degrees of freedom, the descriptor dimensions are not independent for $N > 4$.

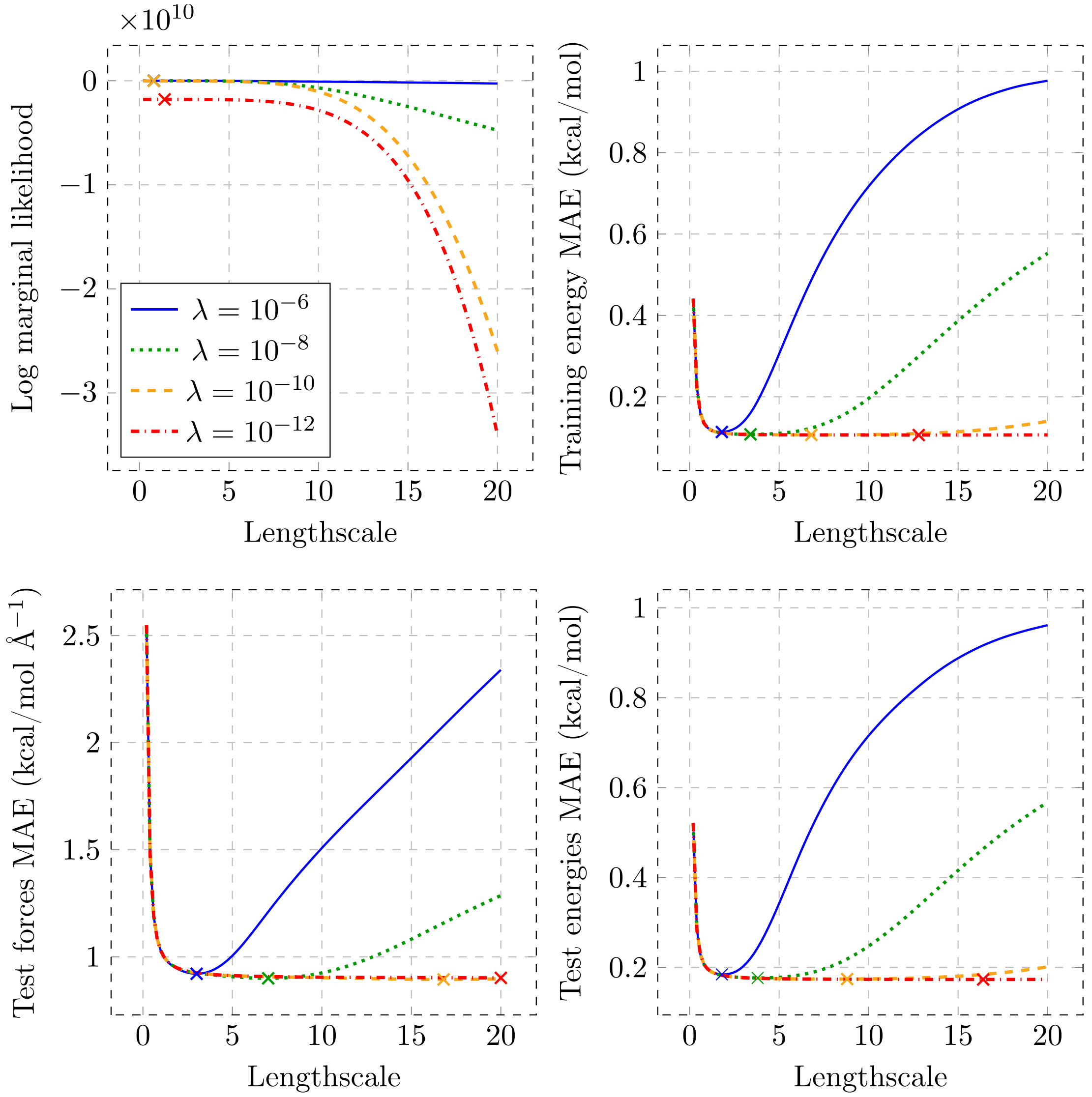

Figure 1. GDML models using the Matérn 5/2 kernel with $l \in [0.2, 20]$ and $\sigma = 1$ trained on 800 and tested on 2000 molecular geometries for ethanol. The maxima of log marginal likelihood (LML) and minima of MAEs for each model are marked with an x. The different curves correspond to different values of λ: $\lambda = 10^{-6}$ (blue), 10^{-8} (green), 10^{-10} (orange), 10^{-12} (red).

Given a descriptor function $D(\boldsymbol{x})$, the kernel function $k_{\mathrm{D}}(\boldsymbol{x}, \boldsymbol{x'}) = k(D(\boldsymbol{x}), D(\boldsymbol{x'}))$ and the Hessian and gradient of the kernel function w.r.t. the input dimensions (as opposed to the descriptor dimensions) become:

$$k_{\mathrm{H}} = J_{\mathrm{D}}(\nabla_{\mathrm{D}} k_{\mathrm{D}} \nabla_{D'}^{T}) J_{D'}^{T} \tag{17}$$

$$k_{\mathrm{G}} = J_{\mathrm{D}} \nabla_{\mathrm{D}} k_{\mathrm{D}} \tag{18}$$

where J_{D} is the Jacobian of the descriptor function w.r.t. the input dimensions. With $D(\boldsymbol{x})$ as inputs, anisotropic kernels have $N(N-1)/2$ parameters with N being the number of atoms. Anisotropic kernels thus built are over-parameterized for $N > 4$.

Kernel models with a large number of parameters cannot be trained by grid search. An alternative to grid search of kernel parameters in kernel ridge regression is maximization of LML in Gaussian process regression. Therefore, we first consider the possibility of building GDML models with Coulomb matrix descriptors by maximizing LML. We begin by analyzing the models with the isotropic Matérn 5/2 kernel at different values of λ over a range of lengthscales l in equation (13) for the molecule ethanol. We sample energies and forces at 800 molecular geometries to generate a training set and at 2000 geometries to generate a hold-out test set. Note that the 800 energies from the training set are not used to train the GDML models (only their mean is used to determine the constant c in equation (10)), as all models in the present work are

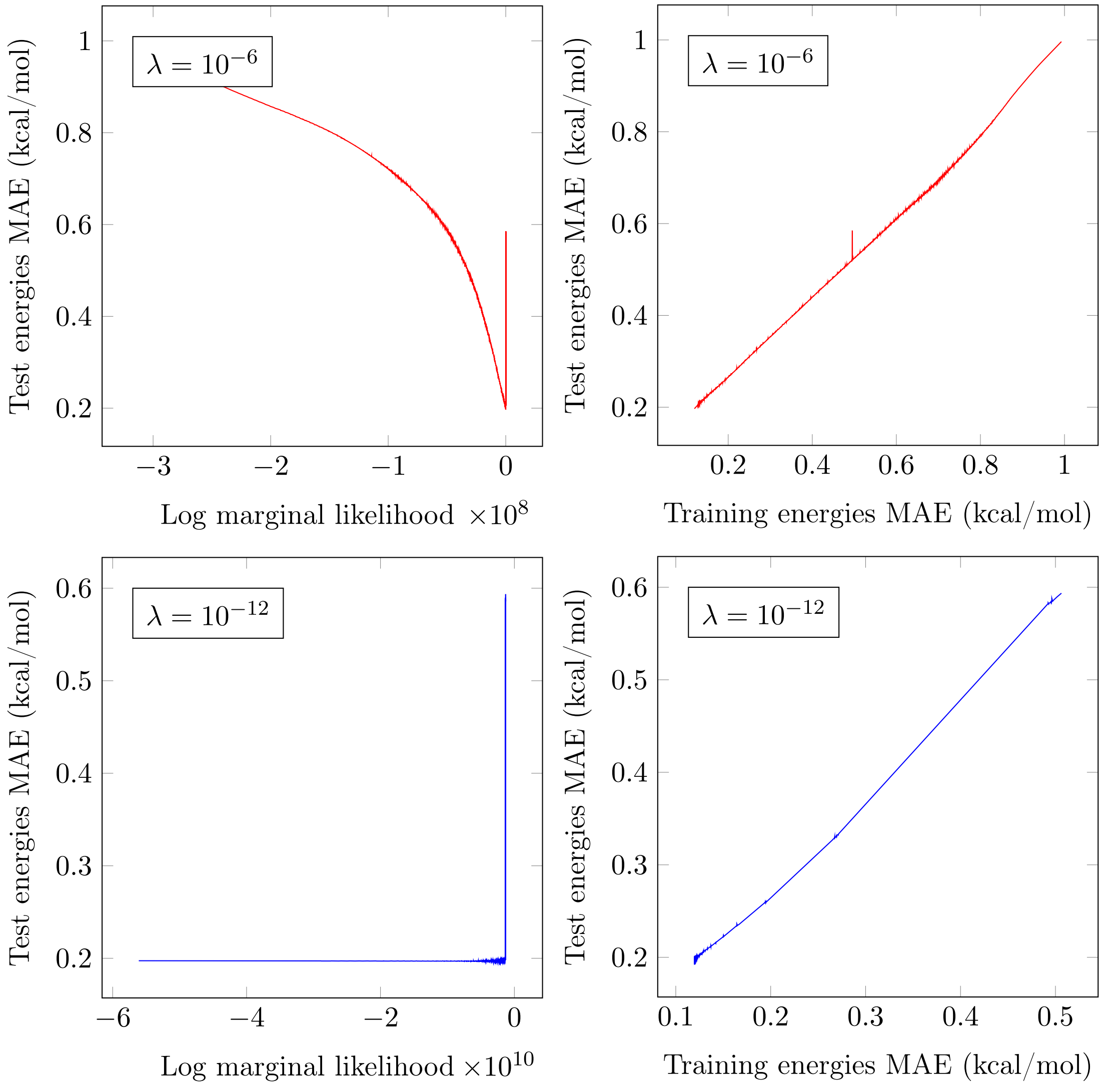

Figure 2. Correlations between LML and test energy errors (left panels) and between training energy errors and test energy errors (right panels) of GDML models with a Matérn 5/2 kernel with $l \in [0.2, 20]$, $\sigma \in [0.5, 10]$, and $\lambda = 10^{-6}$ (upper panels) and $\lambda = 10^{-12}$ (lower panels) trained on 600 and tested on 2000 molecular geometries for ethanol.

trained by forces only. Nevertheless, we refer to these energies as the training energies, as they correspond to the molecular geometries in the training set. Figure 1 displays the LML, MAEs for test energies and test forces and MAEs for training energies. MAEs for training forces are not displayed as the errors of GPs on data used for training are insignificant. Energy values at the training geometries are, however, inferred from the forces and the mean of the training energies, which makes the training energy MAEs for the GDML models in this work similar to the test energy MAEs.

Figures 1 and 2 show that LML cannot be used as a unique model metric for problems with an isotropic Matérn 5/2 kernel (13), as the lowest values of test errors do not correspond to the largest values of LML. The results in figure 1 are obtained with the value $\sigma = 1$, as in previous work [38, 41]. Figure 2 explores the (σ, l) parameter space of the isotropic Matérn 5/2 kernel (13), with $\sigma \in [0.5, 10]$ and $l \in [0.2, 20]$. The calculations are performed for several fixed values of the regularization parameter λ, as indicated in the figures.

We observe that lower values of λ yield lower error values on test data, as should be expected for noiseless problems. However, the LML values for models with small λ are very large and negative which reflects numerical instabilities due to inversion of the unregularized kernel matrix. These numerical instabilities are a consequence of the over-parametrization (i.e. the number of input variables $D(\boldsymbol{x})$ is greater than the number of independent variables). This was observed in [41]. We find similar trends for models with the isotropic RBF and RQ kernels.

Figures 1 and 2 also show that the training energy MAE correlates well with the test energy MAE. Since the energies from the training set are not used for training the GDML models, one can—in principle—use the MAE calculated over energies in the training set for selecting the kernel parameters. To compare the

Table 3. Comparison of GDML models with isotropic simple kernels ($\lambda = 10^{-10}$) trained using LML and training energies with errors computed on 2000 test geometries with the previously published GDML [38] and sGDML [39] results. Best performing GDML models are highlighted in red.

Molecule	Training geometries	Model type	Kernel function	Training method	Test energy MAE (kcal mol^{-1})	Test forces MAE (kcal mol^{-1} Å^{-1})	Reference
Ethanol (21D)	400	GDML	Matérn 5/2	LML	0.280	1.385	This work
				Train E	0.260	1.312	
			RBF	LML	0.256	1.401	
				Train E	0.225	1.191	
			RatQuad	LML	0.237	1.314	
				Train E	**0.225**	**1.191**	
		GDML	Matérn 5/2	Grid search	0.261	1.309	[38]
		sGDML	Matérn 5/2	Grid search	0.103	0.551	[39]

Table 4. Comparison of GDML models with anisotropic simple kernels ($\lambda = 10^{-10}$) trained using LML and training energies with errors computed on 2000 test geometries with the previously published GDML [38] and sGDML [39] results. Best performing GDML models are highlighted in red. The best results with simple isotropic kernels from table 3 are highlighted in blue.

Molecule	Training geometries	Model type	Kernel function	Training method	Test energy MAE (kcal mol^{-1})	Test forces MAE (kcal mol^{-1} Å^{-1})	Reference
Ethanol (21D)	400	AGDML	Matérn 5/2	LML	0.188	0.923	This work
				Train E	0.220	1.144	
			RBF	LML	0.148	0.775	
				Train E	0.187	1.010	
			RatQuad	LML	**0.147**	**0.770**	
				Train E	0.188	1.034	
		GDML	RatQuad	Train E	**0.225**	**1.191**	
		GDML [38]	Matérn 5/2	Grid search	0.261	1.309	[38]
		sGDML [39]	Matérn 5/2	Grid search	0.103	0.551	[39]

efficacy of LML and training energy error for building force-based GDML models with simple kernels, we summarize the most accurate models in table 3. Models labeled 'Train E' are trained by minimizing the training energy RMSE (since MAE has a non-continuous gradient) using a gradient-based optimizer. The results show that models based on training energy error minimization are more accurate than those trained by LML maximization. Table 3 also illustrates that the GDML models with simple kernels are sensitive to the choice of the kernel function.

The results are different for anisotropic kernels. As illustrated in table 4, LML for models with anisotropic kernels becomes the best model metric. Maximization of LML produces the most accurate results. In contrast, RMSE computed over training energies is no longer a good metric. Table 4 also illustrates that the model accuracy is sensitive to the choice of the anisotropic kernel function. Thus, the RQ anisotropic kernel yields significantly better results for both PES and FFs than the models with the Matérn 5/2 kernel. Finally, a comparison of the results in tables 3 and 4 shows that allowing for anisotropy in simple kernels leads to a significant improvement of the models.

We conclude that LML cannot be used for training GDML models with simple isotropic kernels and Coulomb matrix descriptors. The results of figures 1, 2 and table 3 suggest that RMSE over training energies can, potentially, be used to train such models. However, we have found that this is prone to overfitting for problems with anisotropic kernels, making training energy RMSE not suitable for training composite GDML models. As illustrated by table 4, LML can be used for training GMDL models with anisotropic kernels. Since BIC is closely related to LML, one cannot use BIC as a model selection metric for GDML models with isotropic kernels. Kernel anisotropy is, thus, key to the kernel improvement method used in the present work.

3.2. Effect of kernel complexity

We use simple anisotropic kernels as the basis for the kernel construction algorithm described in section 2.3.2. Figures 3 and 4 illustrate the improvement of the AGDML(c) models of both energies and forces for several sets of training geometries as the kernel complexity increases. Figure 3 demonstrates that the maximum values of BIC correspond to the best model at each layer (iteration) of the kernel construction algorithm.

In order to evaluate the effectiveness of the training method based on LML maximization for increasingly more complex kernels, we initialized the kernels with the highest BIC of each layer for 400 and 800 training geometries of ethanol (red dots in figure 3) with random parameters from a log-normal distribution where

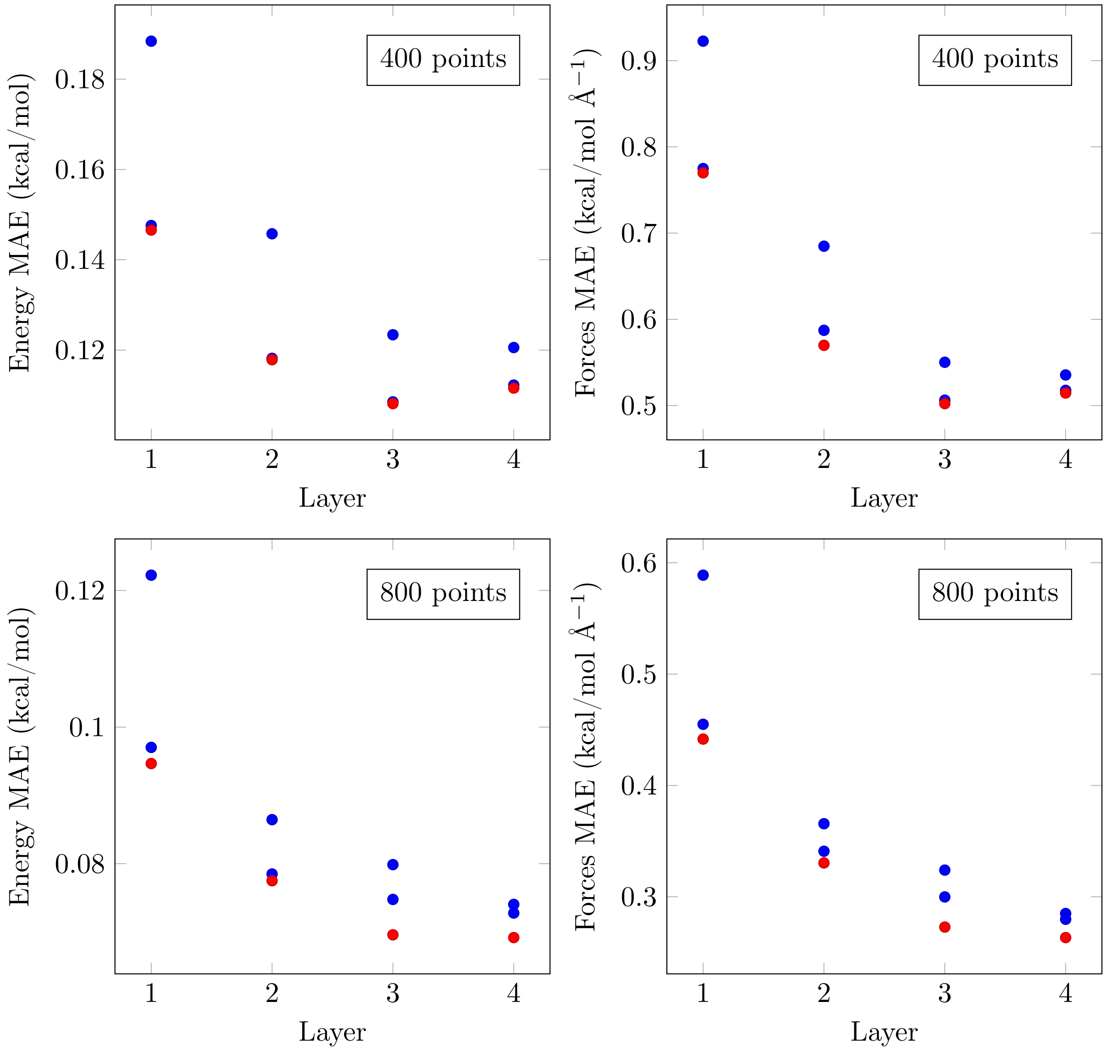

Figure 3. AGDML(c) models for ethanol trained on 400 (upper panels) and 800 (lower panels) molecular geometries by LML maximization and tested on 2000 molecular geometries. Each layer indicates an increase in the number of simple kernel functions in the composite kernel function. The red dots label the models with the maximum value of BIC.

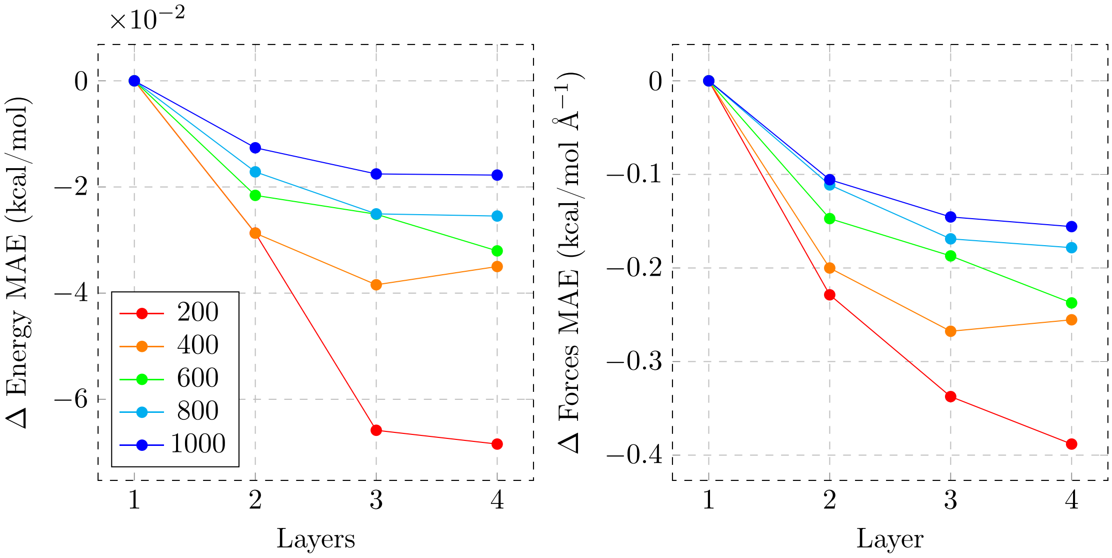

Figure 4. Improvement in the test energy and forces MAEs of AGDML(c) models with increasing kernel complexity for ethanol trained with $n = 200, 400, 600, 800, 1000$ molecular geometries. The errors are computed with 2000 molecular geometries.

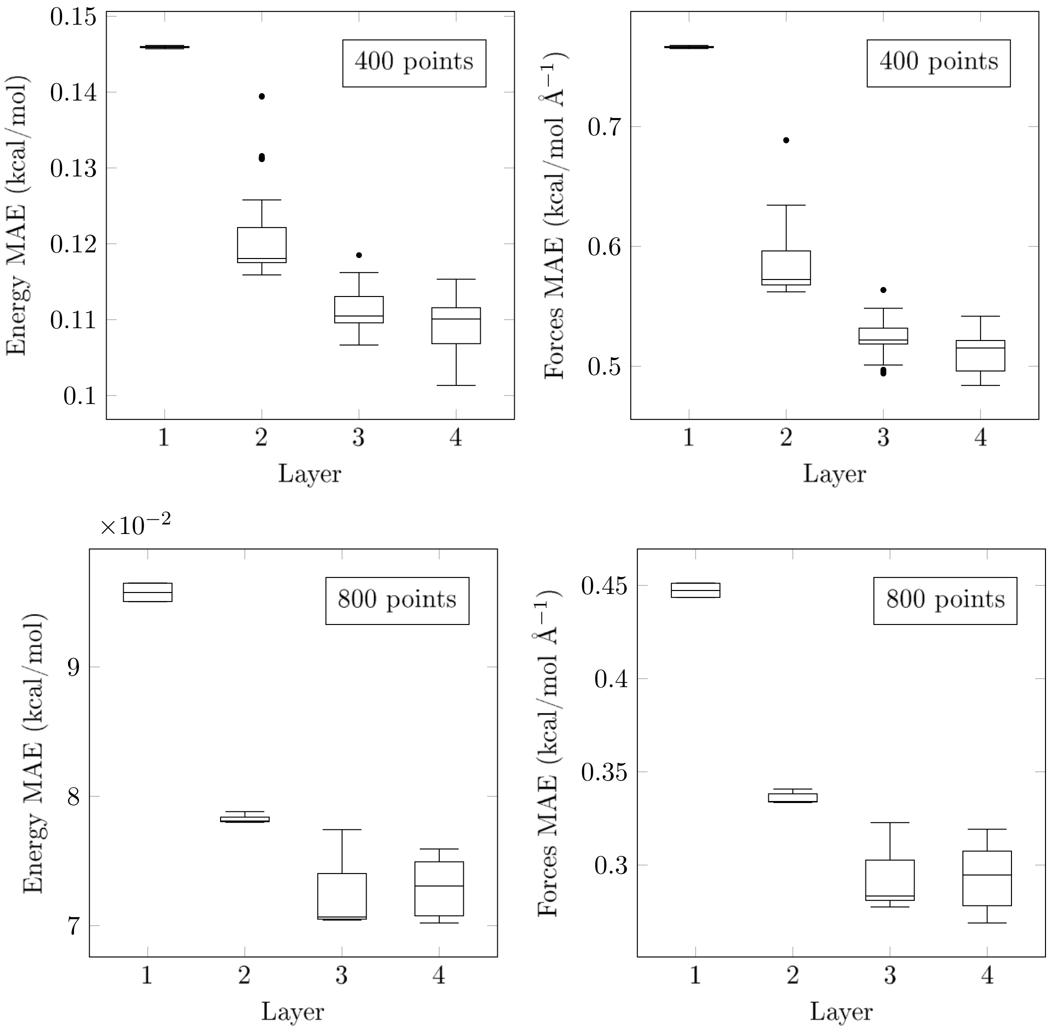

Figure 5. Test energy and forces MAEs for AGDML(c) models for ethanol trained on 400 (upper panels) and 800 (lower panels) molecular geometries by LML maximization and tested on 2000 molecular geometries. Each layer indicates an increase in the number of simple kernel functions in the composite kernel function. The box plots show the distribution of the trained models with all the kernel hyperparameters initialized randomly (40 times for each of the 400 point models and 10 times for each of the 800 point models) from a log-normal distribution where $\log(X) \sim \mathcal{N}(\mu = 0.7, \sigma = 0.2)$.

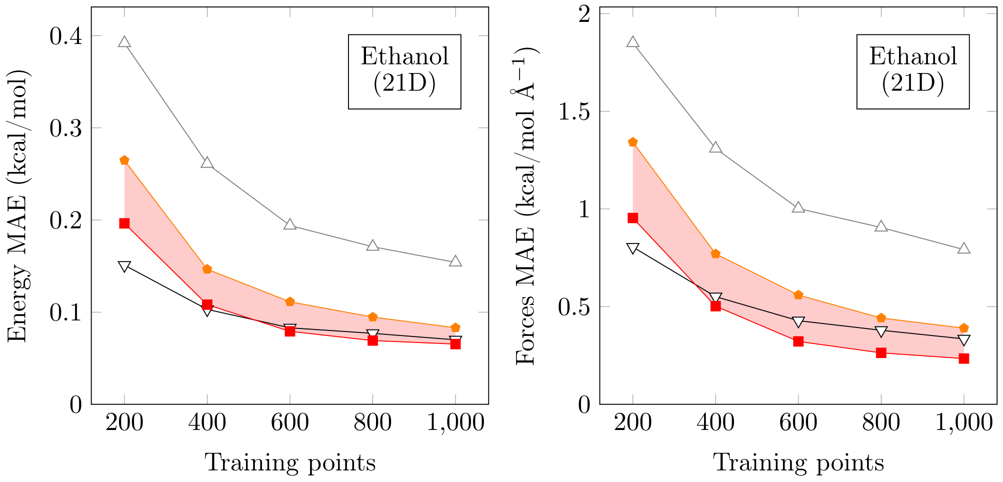

Figure 6. Test energy and forces MAEs for GDML [38] (up triangles), sGDML [39] (down triangles), AGDML (diamonds) and AGDML(c) (squares). The model abbreviations are defined in table 1. The shaded area between the curves indicates the effect of increasing kernel complexity. The test errors are computed with 2000 molecular geometries.

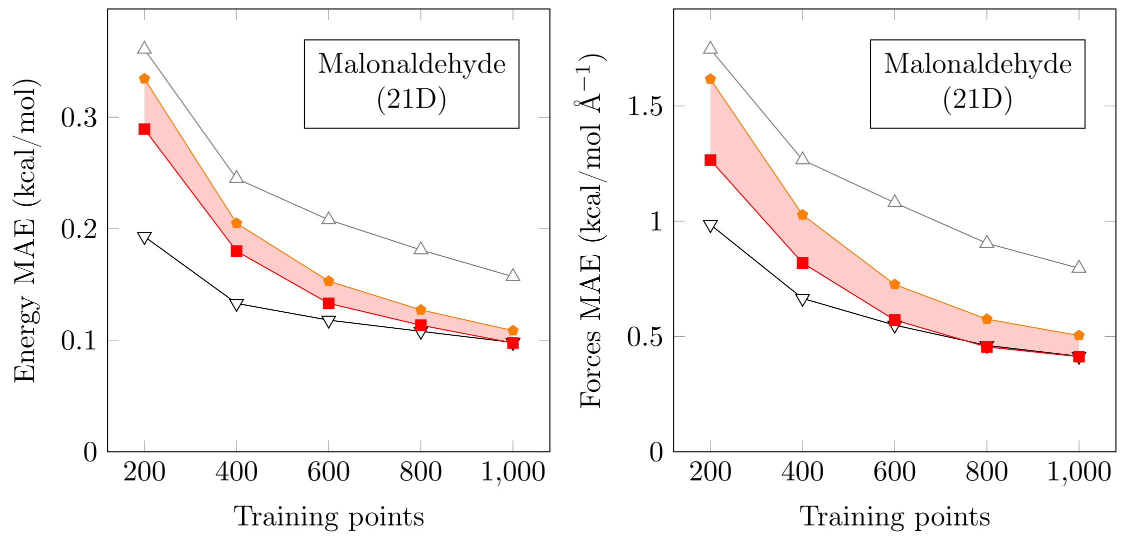

Figure 7. Test energy and forces MAEs for GDML [38] (up triangles), sGDML [39] (down triangles), AGDML (diamonds) and AGDML(c) (squares). The model abbreviations are defined in table 1. The shaded area between the curves indicates the effect of increasing kernel complexity. The test errors are computed with 2000 molecular geometries.

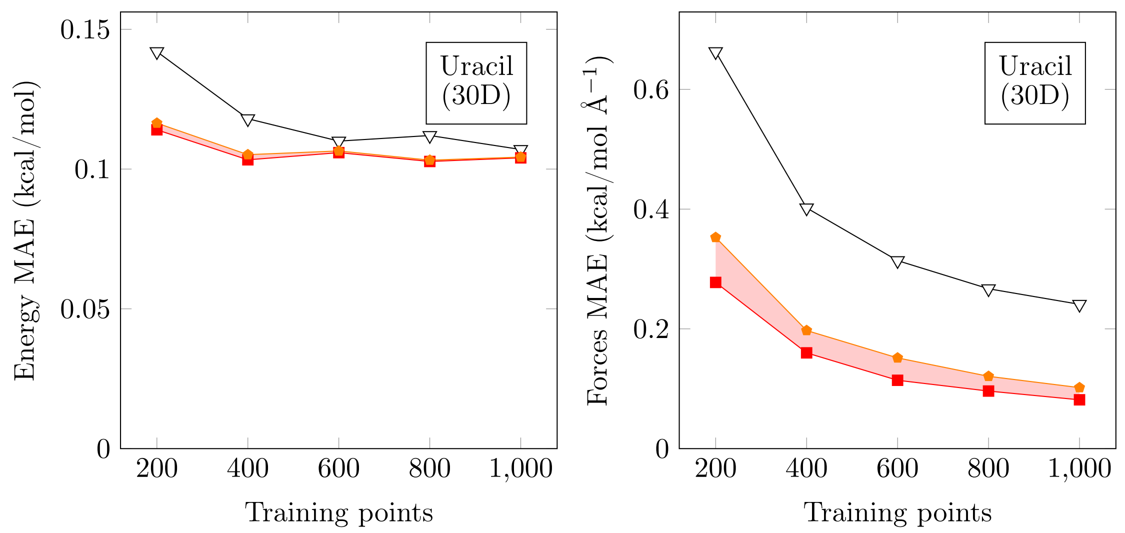

Figure 8. Test energy and forces MAEs for GDML [38] and sGDML [39] (down triangles), AGDML (diamonds) and AGDML(c) (squares). The model abbreviations are defined in table 1. The shaded area between the curves indicates the effect of increasing kernel complexity. The test errors are computed with 2000 molecular geometries.

$\log(X) \sim \mathcal{N}(\mu = 0.7, \sigma = 0.2)$. We found that this distribution of initial hyperparameters produces good results. The corresponding distributions of energy and force MAEs are shown in figure 5 for 40 initializations on the 400 point models and 10 initializations on the 800 point models.

Figures 3 and 5 also show that the accuracy of the models saturates at some iteration of the BIC-driven kernel composition algorithm. Increasing the kernel complexity from layers 3 to 4 results in a decrease of accuracy in figure 3 on 400 points. Each iteration of the composition algorithm adds ∼36 kernel parameters so the number of parameters can become comparable to the number of training geometries as the complexity of the kernel function is increased. As a result, the models can become over-parameterized when trained with a small number of training geometries. This limits the number of layers of kernel complexity that offer a significant improvement of model accuracy. We observe that the model error generally saturates or, in some cases, increases due to overfitting, as the number of kernel layers is further increased.

Figures 6–9 and tables 5–8 display test errors of these models for predictions of both energies and forces for ethanol, malonaldehyde, uracil, and aspirin. In each case, the kernel functions of the AGDML model are chosen based on the values of BIC. The open triangles show the previous results obtained with isotropic Matérn 5/2 kernel functions: the up-triangles represent the GDML results and the down-triangles—the sGDML results. The shaded areas in figures 6–9 show the improvement of the AGDML model accuracy due

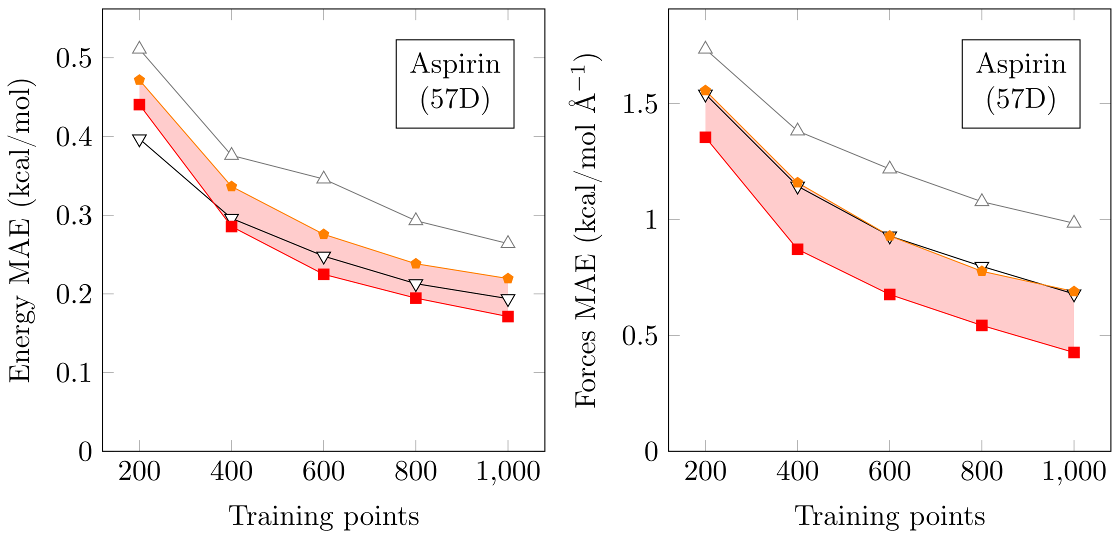

Figure 9. Test energy and forces MAEs for GDML [38] (up triangles), sGDML [39] (down triangles), AGDML (diamonds) and AGDML(c) (squares). The model abbreviations are defined in table 1. The shaded area between the curves indicates the effect of increasing kernel complexity. The test errors are computed with 2000 molecular geometries.

Table 5. Comparison of GDML models with anisotropic composite kernels ($\lambda = 10^{-6}$) trained using LML with errors computed on 2000 test geometries with the previously published GDML [38] and sGDML [39] results. The lowest error values for each number of training geometries is highlighted in red.

Molecule	Training geometries	Model	Test energy MAE (kcal mol^{-1})	Test forces MAE (kcal mol^{-1} Å^{-1})	Reference
Ethanol (21D)	200	AGDML(c) (4 layers)	0.196	0.954	This work
		GDML	0.392	1.849	[38]
		sGDML	**0.151**	**0.805**	[39]
	400	AGDML(c) (4 layers)	0.108	**0.502**	This work
		GDML	0.261	1.309	[38]
		sGDML	**0.103**	0.551	[39]
	600	AGDML(c) (4 layers)	**0.0792**	**0.322**	This work
		GDML	0.194	1.002	[38]
		sGDML	0.083	0.428	[39]
	800	AGDML(c) (4 layers)	**0.0692**	**0.263**	This work
		GDML	0.171	0.905	[38]
		sGDML	0.077	0.379	[39]
	1000	AGDML(c) (4 layers)	**0.0653**	**0.234**	This work
		GDML	0.154	0.792	[38]
		sGDML	0.072	0.335	[39]

to increasing the complexity of the kernel functions. The upper edges of the bands (diamonds) depict the AGDML results with simple kernels, whereas the lower edges (squares) correspond to the AGDML results with composite kernels. As uracil does not have any permutational symmetry, the reference GDML and sGDML models in figure 8 have the same test error values and are displayed together as a single curve. We observe that the improvement due to the increasing complexity of GDML kernels is more significant for the prediction of forces and for larger molecules.

As evident from figures 6–9, the accuracy of the AGDML(c) models is comparable with and, in some cases, better than the accuracy of the sGDML models. Building symmetries into kernel functions is a molecule-specific task. In contrast, the kernel construction algorithm presented here is completely general and easy to automate. In addition, the method illustrated here can be applied to sGDML calculations, further improving the sGDML results.

Table 6. Comparison of GDML models with anisotropic composite kernels ($\lambda = 10^{-6}$) trained using LML with errors computed on 2000 test geometries with the previously published GDML [38] and sGDML [39] results. The lowest error values for each number of training geometries is highlighted in red.

Molecule	Training geometries	Model	Test energy MAE (kcal mol^{-1})	Test forces MAE (kcal mol^{-1} Å^{-1})	Reference
Malonaldehyde (21D)	200	AGDML(c) (3 layers)	0.290	1.265	This work
		GDML	0.361	1.746	[38]
		sGDML	**0.193**	**0.985**	[39]
	400	AGDML(c) (3 layers)	0.180	0.819	This work
		GDML	0.245	1.266	[38]
		sGDML	**0.133**	**0.665**	[39]
	600	AGDML(c) (3 layers)	0.133	0.571	This work
		GDML	0.208	1.080	[38]
		sGDML	**0.118**	**0.549**	[39]
	800	AGDML(c) (3 layers)	0.113	**0.454**	This work
		GDML	0.181	0.904	[38]
		sGDML	**0.108**	0.461	[39]
	1000	AGDML(c) (3 layers)	**0.098**	**0.389**	This work
		GDML	0.157	0.796	[38]
		sGDML	0.098	0.414	[39]

Table 7. Comparison of GDML models with anisotropic composite kernels ($\lambda = 10^{-6}$) trained using LML with errors computed on 2000 test geometries with the previously published GDML [38] and sGDML [39] results. The lowest error values for each number of training geometries is highlighted in red.

Molecule	Training geometries	Model	Test energy MAE (kcal mol^{-1})	Test forces MAE (kcal mol^{-1} Å^{-1})	Reference
Uracil (30D)	200	AGDML(c) (4 layers)	**0.114**	**0.278**	This work
		(s)GDML	0.142	0.663	[38, 39]
	400	AGDML(c) (4 layers)	**0.103**	**0.163**	This work
		(s)GDML	0.118	0.402	[38, 39]
	600	AGDML(c) (4 layers)	**0.106**	**0.114**	This work
		(s)GDML	0.110	0.314	[38, 39]
	800	AGDML(c) (3 layers)	**0.103**	**0.096**	This work
		(s)GDML	0.112	0.267	[38, 39]
	1000	AGDML(c) (3 layers)	**0.104**	**0.082**	This work
		(s)GDML	0.107	0.241	[38, 39]

As indicated by previous work [29] with low dimensional molecules, the improvement due to composite kernels is more pronounced for smaller numbers of training points. However, models of large molecules with anisotropic kernels require a large number of kernel parameters. We observe that even for large molecules, models with composite kernels trained by a small number of training geometries are stable and generalize well. An interesting case to examine is the model of aspirin with anisotropic composite kernels. The number of training parameters added at each layer of the kernel construction algorithm of section 2.3.2 for the AGDML(c) model for aspirin is ≈210. This leads to models with ≈420 trainable parameters for kernels with two layers of complexity and ≈630 trainable parameters for kernels with three layers of complexity used to obtain results in figure 9. In such models, the number of composite kernel parameters can be larger than the number of training points. This is not unusual in ML and occurs commonly in models based on complex neural networks. While such over-parameterized models are ill-defined from a statistical point of view, empirically they generalize well [54]. We observe that models of aspirin based on composite kernels with ≈630 parameters generalize better than models with simple anisotropic kernels, even when trained by $\leqslant$600 training geometries.

Table 8. Comparison of GDML models with anisotropic composite kernels ($\lambda = 10^{-6}$) trained using LML with errors computed on 2000 test geometries with the previously published GDML [38] and sGDML [39] results. The lowest error values for each number of training geometries is highlighted in red.

	Training		Test energy MAE	Test forces MAE	
Molecule	geometries	Model	(kcal mol^{-1})	(kcal mol^{-1} Å^{-1})	Reference
Aspirin (57D)	200	AGDML(c) (3 layers)	0.441	**1.355**	This work
		GDML	0.511	1.735	[38]
		sGDML	**0.397**	1.541	[39]
	400	AGDML(c) (3 layers)	**0.286**	**0.872**	This work
		GDML	0.376	1.382	[38]
		sGDML	0.296	1.144	[39]
	600	AGDML(c) (2 layers)	**0.225**	**0.677**	This work
		GDML	0.346	1.218	[38]
		sGDML	0.248	0.929	[39]
	800	AGDML(c) (2 layers)	**0.195**	**0.543**	This work
		GDML	0.293	1.077	[38]
		sGDML	0.213	0.798	[39]
	1000	AGDML(c) (2 layers)	**0.177**	**0.457**	This work
		GDML	0.264	0.984	[38]
		sGDML	0.194	0.679	[39]

4. Conclusions

The generalization accuracy of ML models of PES and FFs for large polyatomic molecules can be typically improved by: (a) increasing the number of training points; (b) improving the descriptors; (c) improving the model. In order to build accurate models based on *ab initio* calculations, much of recent work has focused on (b) and (c). In particular, it has been shown that GDML models produce accurate results for high-dimensional molecular systems with a small number of *ab initio* calculations. The present work illustrates that GDML models can be further improved by increasing the complexity of underlying kernels through a greedy search algorithm with BIC as model selection. We have shown that this requires allowing for kernel anisotropy and produces significantly improved results. For example, we show that the GDML model trained by 1000 *ab initio* calculations produces a global 57-dimensional PES for aspirin with the mean absolute accuracy 0.177 kcal mol^{-1} (61.9 cm^{-1}) and FFs with the mean absolute accuracy 0.457 kcal mol^{-1} Å^{-1}. This requires composite kernels with 424 trainable parameters. We emphasize that such kernels cannot be chosen at random. As illustrated in [51], the iterative kernel construction algorithm based on BIC optimization is critical for building models with low generalization error.

We note that the method demonstrated here can be used with either GDML or sGDML models. sGDML models take advantage of molecular symmetries to reduce the size of the input space, which results in better accuracy with the same number of training points. The models presented here could also be improved by expanding the set of simple kernels to include more functions. This can increase the space of kernels, potentially offering more flexibility for the kernel optimization algorithm. In the present work, we have only included kernel functions that are doubly differentiable. It would be interesting to explore an approach that is based on combinations of doubly and singly differentiable kernel functions. An example of a suitable singly differentiable function is a Matérn 3/2 function. The latter could be used to improve the accuracy of the PES models without affecting the models of FFs. The kernel optimization algorithm could further be improved by including products as well as linear combinations of kernels. This would complicate the computation of the Hessians but is not a fundamental obstacle. Finally, it would be interesting to include non-stationary kernels into the set of basis kernel functions. Non-stationary kernels have been proven important for extrapolation problems explored in previous studies [28, 45]. We have not studied the extrapolation properties of AGDML(c) models. Extrapolation outside the coordinate space of the training data is unlikely to produce accurate results with the current set of kernel functions that are all stationary. Previous studies using GPs for extrapolation [28, 45] have shown promising results predicting energies outside the range of energies in the training data. We expect GDML and AGDML(c) to also be effective at extrapolating in the energy or forces domain, but more in depth studies are needed.

Our present work and the potential for further improvements thus indicate that it is possible to construct high-dimensional PES and FFs for large polyatomic molecules with accuracy exceeding 0.1 kcal mol^{-1} with a remarkably small number of quantum chemistry calculations ($<$1000 for molecules with $>$50 degrees of freedom). The kernel construction algorithm of the present work does not use any prior knowledge of the PES or FF landscape. The models are trained by forces at randomly chosen molecular geometries. There is no need for sophisticated sampling schemes such as ones based on active learning that usually require a significantly larger number of *ab initio* calculations than random sampling. The present approach is therefore readily applicable to any molecular systems.

Data availability statement

The data that support the findings of this study are openly available at the following URL: http://quantum-machine.org/gdml/.

Acknowledgment

This work was supported by NSERC of Canada.

ORCID iD

K Asnaashari https://orcid.org/0000-0001-5176-3153

References

[1] Schütt K T, Arbabzadah F, Chmiela S, Müller K R and Tkatchenko A 2017 *Nat. Commun.* **8** 13890
[2] Unke O T and Meuwly M 2019 *J. Chem. Theory Comput.* **15** 3678
[3] Manzhos S and Carrington T J 2006 *J. Chem. Phys.* **125** 084109
[4] Manzhos S, Wang X, Dawes R and Carrington T J 2006 *J. Phys. Chem.* A **110** 5295
[5] Behler J and Parrinello M 2007 *Phys. Rev. Lett.* **98** 146401
[6] Behler J 2011 *Phys. Chem. Chem. Phys.* **13** 17930
[7] Behler J 2015 *Int. J. Quantum Chem.* **115** 1032
[8] Pradhan E and Brown A 2017 *Phys. Chem. Chem. Phys.* **19** 22272
[9] Leclerc A and Carrington T J 2014 *J. Chem. Phys.* **140** 174111
[10] Manzhos S, Dawes R and Carrington T 2015 *Int. J. Quantum Chem.* **115** 1012
[11] Chen J, Xu X, Xu X and Zhang D H 2013 *J. Chem. Phys.* **138** 154301
[12] Liu Q, Zhou X, Zhou L, Zhang Y, Luo X, Guo H and Jiang B 2018 *J. Phys. Chem.* C **122** 1761
[13] Manzhos S and Carrington T 2021 *Chem. Rev.* **121** 10187
[14] Meuwly M 2021 *Chem. Rev.* **121** 10218
[15] Handley C M, Hawe G I, Kellab D B and Popelier P L A 2009 *Phys. Chem. Chem. Phys.* **11** 6365
[16] Bartók A P, Payne M C, Kondor R and Csányi G 2010 *Phys. Rev. Lett.* **104** 136403
[17] Bartók A P and Csányi G 2015 *Int. J. Quantum Chem.* **115** 1051
[18] Cui J and Krems R V 2016 *J. Phys. B: At. Mol. Opt. Phys.* **49** 224001
[19] Dral P O, Owens A, Yurchenko S N and Thiel W 2017 *J. Chem. Phys.* **146** 244108
[20] Kolb B, Marshall P, Zhao B, Jiang B and Guo H 2017 *J. Phys. Chem.* A **121** 2552
[21] Kamath A, Vargas-Hernandez R A, Krems R V, Carrington T J and Manzhos S 2018 *J. Chem. Phys.* **148** 241702
[22] Schmitz G and Christiansen O 2018 *J. Chem. Phys.* **148** 241704
[23] Guan Y, Yang S and Zhang D H 2018 *Mol. Phys.* **116** 823
[24] Laude G, Calderini D, Tew D P and Richardson J O 2018 *Faraday Discuss.* **212** 237
[25] Guan Y, Yang S and Zhang D H 2018 *J. Phys. Chem.* A **122** 3140
[26] Wiens A E, Copan A V and Schaefer H F 2019 *Chem. Phys. Lett.* **3** 100022
[27] Qu C, Yu Q, Van Hoozen B L J, Bowman J M and Vargas-Hernàndez R A 2018 *J. Chem. Theory Comput.* **14** 3381
[28] Sugisawa H, Ida T and Krems R V 2020 *J. Chem. Phys.* **153** 114101
[29] Dai J and Krems R V 2020 *J. Chem. Theory Comput.* **16** 1386–95
[30] Unke O T and Meuwly M 2017 *J. Chem. Inf. Model.* **57** 1923
[31] Ho T S and Rabitz H 1996 *J. Chem. Phys.* **104** 2584
[32] Hollebeek T, Ho T S and Rabitz H 1997 *J. Chem. Phys.* **106** 7223
[33] Ho T S and Rabitz H 2003 *J. Chem. Phys.* **119** 6433
[34] Unke O T, Chmiela S, Sauceda H E, Gastegger M, Poltavsky I, Schütt K T, Tkatchenko A and Müller K-R 2021 *Chem. Rev.* **121** 10142
[35] Carleo G, Cirac I, Cranmer K, Daudet L, Schuld M, Tishby N, Vogt-Maranto L and Zdeborová L 2019 *Rev. Mod. Phys.* **91** 045002
[36] Krems R V 2019 *Phys. Chem. Chem. Phys.* **21** 13392
[37] Behler J 2016 *J. Chem. Phys.* **145** 170901
[38] Chmiela S, Tkatchenko A, Sauceda H E, Poltavsky I, Schütt K T and Müller K R 2017 *Sci. Adv.* **3** 1603015
[39] Chmiela S, Sauceda H E, Müller K R and Tkatchenko A 2018 *Nat. Commun.* **9** 3887
[40] Chmiela S, Sauceda H E, Poltavsky I, Müller K R and Tkatchenko A 2019 *Comput. Phys. Commun.* **240** 38
[41] Chmiela S 2019 *Doctoral Thesis* Technische Universität Berlin
[42] Faber F A, Christensen A S, Huang B and von Lilienfeld O A 2018 *J. Chem. Phys.* **148** 241717
[43] Christensen A S, Bratholm L A, Faber F A and von Lilienfeld O A 2020 *J. Chem. Phys.* **152** 044107

[44] Duvenaud D K, Nickisch H and Rasmussen C E 2011 *Advances in Neural Information Processing Systems* ed Shawe-Taylor J, Zemel R, Bartlett P, Pereira F and Weinberger K Q vol 24 p 226

[45] Duvenaud D, Lloyd J, Grosse R, Tenenbaum J and Zoubin G 2013 *Proc. 30th Int. Conf. on Machine Learning* vol 28, ed S Dasgupta and D McAllester (Atlanta, GA: PMLR) pp 1166–74

[46] Vargas-Hernàndez R A, Sous J, Berciu M and Krems R V 2018 *Phys. Rev. Lett.* **121** 255702

[47] Wilson A and Adams R 2013 *Proc. 30th Int. Conf. on Machine Learning* vol 28, ed S Dasgupta and D McAllester (Atlanta, GA: PMLR) pp 1067–75

[48] Lázaro-Gredilla M, Quiñnero-Candela J, Rasmussen C E and Figueiras-Vidal A R 2010 *J. Mach. Learn. Res.* **11** 1865–81

[49] Remes S, Heinonen M and Kaski S 2017 *J. Mach. Learn. Res.* **30** 4642

[50] Solak E, Murray-Smith R, Leithead W E, Leith D J and Rasmussen C E 2002 *Proc. 15th Int. Conf. on Neural Information Processing Systems, NIPS'02* (Cambridge, MA: MIT Press) pp 1057–64

[51] Vargas-Hernández R A and Krems R V 2020 *Machine Learning Meets Quantum Physics* vol 968, ed K T Schütt, S Chmiela, O A von Lilienfeld, A Tkatchenko, K Tsuda and K-R Müller (Cham: Springer) pp 171–94

[52] Perdew J P, Burke K and Ernzerhof M 1996 *Phys. Rev. Lett.* **77** 3865–8

[53] Tkatchenko A and Scheffler M 2009 *Phys. Rev. Lett.* **102** 073005

[54] Canatar A, Bordelon B and Pehlevan C 2021 *Nat. Commun.* **12** 2914