The Journal of Chemical Physics

Improving molecular force fields across configurational space by combining supervised and unsupervised machine learning

Cite as: J. Chem. Phys. **154**, 124102 (2021); https://doi.org/10.1063/5.0035530
Submitted: 29 October 2020 • Accepted: 01 March 2021 • Published Online: 22 March 2021

Gregory Fonseca, Igor Poltavsky, Valentin Vassilev-Galindo, et al.

The Journal of Chemical Physics ARTICLE scitation.org/journal/jcp

Improving molecular force fields across configurational space by combining supervised and unsupervised machine learning

Cite as: J. Chem. Phys. 154, 124102 (2021); doi: 10.1063/5.0035530
Submitted: 29 October 2020 • Accepted: 1 March 2021 •
Published Online: 22 March 2021

Gregory Fonseca, Igor Poltavsky, Valentin Vassilev-Galindo, and Alexandre Tkatchenko[a)]

AFFILIATIONS

Department of Physics and Materials Science, University of Luxembourg, L-1511 Luxembourg, Luxembourg

[a)]Author to whom correspondence should be addressed: alexandre.tkatchenko@uni.lu

ABSTRACT

The training set of atomic configurations is key to the performance of any Machine Learning Force Field (MLFF) and, as such, the training set selection determines the applicability of the MLFF model for predictive molecular simulations. However, most atomistic reference datasets are inhomogeneously distributed across configurational space (CS), and thus, choosing the training set randomly or according to the probability distribution of the data leads to models whose accuracy is mainly defined by the most common close-to-equilibrium configurations in the reference data. In this work, we combine unsupervised and supervised ML methods to bypass the inherent bias of the data for common configurations, effectively widening the applicability range of the MLFF to the fullest capabilities of the dataset. To achieve this goal, we first cluster the CS into subregions similar in terms of geometry and energetics. We iteratively test a given MLFF performance on each subregion and fill the training set of the model with the representatives of the most inaccurate parts of the CS. The proposed approach has been applied to a set of small organic molecules and alanine tetrapeptide, demonstrating an up to twofold decrease in the root mean squared errors for force predictions on non-equilibrium geometries of these molecules. Furthermore, our ML models demonstrate superior stability over the default training approaches, allowing reliable study of processes involving highly out-of-equilibrium molecular configurations. These results hold for both kernel-based methods (sGDML and GAP/SOAP models) and deep neural networks (SchNet model).

I. INTRODUCTION

With the enormous rise in computational power and the number of molecular simulation methods in the past decades, atomistic modeling is increasingly becoming the method of choice.[1–9] Applications range from the study and prevention of corrosion[10–12] to protein folding,[13] unfolding,[14] and self-assembly.[15]

For many applications, machine learning force fields (MLFFs) are becoming the method of choice, as they can potentially reproduce any functional form of interatomic and intermolecular interactions, leading to reliable descriptions of potential energy surfaces (PESs) of arbitrary complexity. Many successes have been found in this domain in the recent years, with a multitude of methods being able to predict the behavior of small to medium sized molecules and more.[16–23] These methods were used to calculate the stability of molecules with chemical accuracy,[24] predict the formation energy of crystals at the level of density functional theory,[25] or even reconstruct phase diagrams,[26] to name a few examples.

Despite these achievements, the data-driven nature of ML has its downsides: collecting data and choosing training points is a nontrivial problem that requires a deep understanding of the nature of the data, which relies on human intuition. This puts into question the unbiased nature of the ML approaches, eliminating one of their main advantages over the human-designed FFs. For instance, for applications in molecular dynamics (MD) simulations, the training data are generally parts of molecular trajectories extracted from a reference *ab initio* simulation with the desired level of accuracy. ML models are then frequently trained to have the best overall prediction across the entire dataset. This, however, skews the ML models toward more common (close-to-equilibrium) molecular configurations, as poorly predicted but rare (out-of-equilibrium) configurations hardly impact overall statistics.

The performance of such ML models can be unpredictable in long MD simulations where small regions of configurational space (CS) with poor predictions can act as an escape way toward the extrapolation regime. Specific examples include studying nuclear quantum effects (such as proton transport) using the MLFF trained on classical MD trajectories or simulating phase transitions based on the information collected only in stable phases. In all these cases, minimizing the prediction error on the reference dataset does not guarantee good predictions across all important configurations, making the results of the simulations questionable.

In this work, we address the issue outlined above by "flattening the error" of ML models: i.e., we ensure that the predictive accuracy of the MLFF is equally reliable for out-of-equilibrium structures or rare events as for common configurations, thus enhancing the stability of the model regardless of its use case. To accomplish this, we propose a novel method to optimize the training of ML models, leading to unbiased molecular FFs with almost constant accuracy across the entire reference dataset. This method is equally applicable to any ML model and is available in our free open-source MLFF package.[27] We showcase its application on small organic molecules (uracil, salicylic acid, ethanol, and toluene) as well as a larger molecule (alanine tetrapeptide) using GAP[28] models with the SOAP[29] descriptor and sGDML[17] as representative kernel-based approaches, and SchNet[18] as a representative neural-network-based approach. Those models were chosen in order to cover a large variety of ML methods (kernel with cutoff, kernel without cutoff, neural network); it is important to note, however, that the exact model choice is not important as our methods do not affect the model itself but merely influence its training set. Comparing our improved models to default ones of equal training set size reveals an error reduction on rare/out-of-equilibrium configurations by a factor of up to 2 for a negligible sacrifice in mean error. The proposed training scheme improves the reliability of the ML models, substantially widening their application range. Furthermore, our methods highlight how common metrics to determine the accuracy of an ML model (e.g., root-mean-squared errors on a test set) are incomplete and unable to capture the true reliability of the model.

This article is organized as follows: in Sec. II, we explain and justify the importance of our work in the current state of research. Section III contains the details of the developed methodology for outlier detection and improved training technique. In Sec. IV, we apply our method to reconstruct the FF of small organic molecules and alanine tetrapeptide, which serves as a representative case for how the method performs on larger molecules. Section V presents a summary and an outlook.

II. MOTIVATION

At the core of every ML model lie the reference data. In the case of reproducing molecular FFs, generating reliable data is quite challenging, and thus, many different sampling techniques exist (e.g., MD and normal mode sampling), with a lot of work being done to improve them.[30–34] In practice, we are ultimately always forced to generate large amounts of data using quantum chemical methods that are as cheap as possible while still qualitatively reproducing the underlying physics. However, as training sets of ML models are usually significantly smaller than the reference datasets, the points inside them can be recalculated using higher-accuracy quantum chemistry methods. This effectively increases the accuracy of the MLFF model to that of the chosen method without the need to recompute the entire reference data. Nevertheless, this still represents the most computationally expensive step in the process of MLFF construction solely due to the excessive costs associated with highly accurate quantum-mechanical methods.

This motivates the creation of data-efficient MLFFs, i.e., models able to reproduce the PES of a molecule with a training set as small as possible. In that setting, the specific choice of training points becomes essential. In particular, this promotes avoiding redundant points in the training set and ensuring that all CS regions visited during the generation of the reference data have adequate representation in the training set. Performing the selection of training points randomly or along the reference data's inherent distributions (for example energy) does not guarantee that those conditions are met. The methods presented in this work aim to create the best possible training set for a given reference dataset and a given ML model.

It is important to note that our methods only change the choice of training points and are, in principle, independent of the choice of the ML model, the tuning of its hyperparameters, and its computational costs for prediction. As a result, our method can be used as a complementary approach to other ML techniques at an initial, intermediate, or final step of training highly accurate and reliable ML models. The proposed technique leads to increased training time (usually by a factor of 4–5), as the chosen model will need to be trained multiple times on a training set with iteratively increasing size. However, the computational costs of the training process are still very small compared to that of recalculating the training set using accurate quantum-mechanical methods.

For large enough training sets, any reliable ML model should, in principle, be able to reproduce the PES with arbitrary accuracy. In contrast, in the limit of small datasets, simple interpolation between reference points is not enough for accurate predictions; instead, models have to learn the underlying physics. Our method is able to find an optimal training set of a given size for each dataset and model combination. This allows even small training sets to fully and accurately represent the PES. As such, we can now compare the inherent ability of a given MLFF model to learn complex interatomic forces. Thus, the developed method allows us to assess the expressive power of ML models for reproducing molecular PESs in an unbiased way.

III. METHODS

A. Overview

The methods described in this section allow for an in-depth error analysis of a ML model, outlier detection, as well as an improved training technique resulting in "equally" reliable predictions for all parts of the entire reference dataset. We apply the developed approach for constructing accurate MLFFs for molecules consisting of a few tens of atoms, but it can be generalized for any regression problem. An overview of the methodology can be found in Fig. 1.

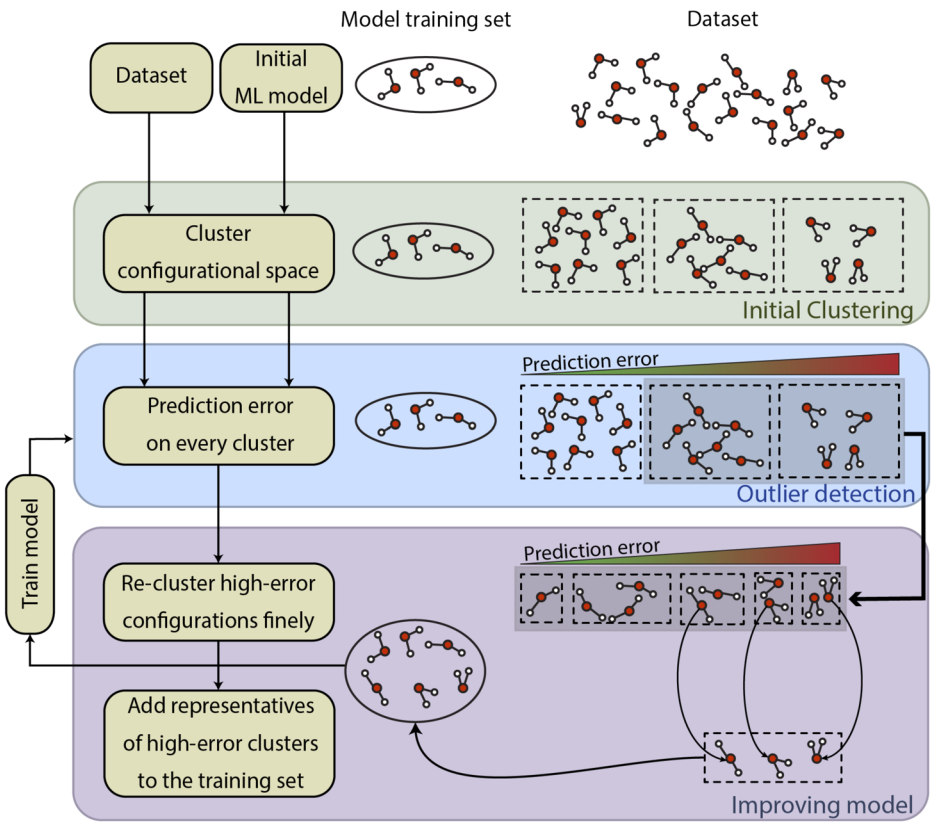

FIG. 1. Overview of the improved learning method. A dataset is clustered into subsets, and the error of an initial ML model is assessed on each individual cluster. High-error clusters are reclustered finely, and representative configurations are extracted from each and added to the training set. This is repeated until a given number of training points is reached.

The method can be subdivided into three main steps. In the "initial clustering" step, we split molecular configurations into groups based on the similarities in geometric and energetic properties. We then apply a given ML model to each individual subset and compute the respective mean prediction error, identifying the worst predicted subsets as "outliers." Finally, to create "improved models," we more finely subdivide poorly predicted groups and extract representative geometries from each fine cluster. These representatives are added to the training set. Repeating the described procedure with retrained models results in a final model with an optimized training set. The details of every step are in the following.

B. Details

In the "initial clustering" step, we subdivide tens of thousands of unlabeled data points into 50 initial clusters. In the general case, the number of initial clusters is an adjustable parameter requiring some intuition about the complexity of the PES and the knowledge about the reference dataset size. The specific number should be adjusted to create clusters broad enough to represent subregions of CS but not so large to mix largely different configurations. For the datasets and molecules used in this work, 50 was found to be a reliable value, meaning that a specific choice of this value allows some flexibility, and one should not optimize it too much for every single problem.

In this paper, pairwise atomic distances are the descriptor of choice for molecular geometries, and differences between configurations were defined using the Euclidean distance in this descriptor space. For the applications in this paper, molecules are small enough for such a simple metric to adequately represent the configurational space. An agglomerative approach with Ward linkage[35] was chosen to cluster the dataset into configurations with similar geometries (ten clusters), as the algorithm avoids merging rare but geometrically unique configurations with large groups of common ones. Note that this clustering method has a time complexity of $O(n^3)$ and memory requirement of $O(n^2)$ (with n being the number of points to cluster), making it inadequate for handling large datasets in one go; instead, in a first step, a subset of the data (20 000 points) is chosen and clustered. The remaining (unclustered) data points are then iteratively added to an existing cluster based on the smallest average distance, mimicking agglomerative clustering while bypassing computational limitations.

With $\vec{x}_i^a$ being the Euclidean position of atom $i \in [1, N]$ of data point $a \in [1, M]$, the descriptor $\vec{z}^a$ is given by

$$\vec{z}^a = [\ldots, z^a_{i,j}, \ldots], \; j < i, \tag{1}$$

$$z^a_{i,j} = \|\vec{x}_i^a - \vec{x}_j^a\|_2, \tag{2}$$

with $\|\cdot\|_2$ being a simple Euclidean distance. Distances in the descriptor space are then

$$d(\vec{z^a}, \vec{z^b}) = \|\vec{z^a} - \vec{z^b}\|_2. \quad (3)$$

Despite the similar geometry, clusters produced this way may contain large variations in the potential energy. To avoid this problem, a further distinction between different energy levels was done by further splitting each cluster into five using a KMeans method with kmeans++ initialization[36,37] on the potential energies. The combination of both clustering techniques helped distinguish between possible degenerate states and geometrically "similar" configurations with significant energy differences. All clustering algorithms in this work were implemented using the Scikit-learn library.[38]

After successfully splitting the dataset both by geometries and energies, an initial ML model was applied to every individual cluster—the "outlier detection" step. The training set of the initial model should be small enough to allow for further training points to be added but large enough for predictions to be qualitatively correct. For this work, initial training set sizes were 20% that of the final improved models (i.e., 200 training points). The choice of points for this set was left unaltered from the default methods used by respective ML models.

The average prediction error on all configurations was computed between the predicted and actual forces. The root-mean-squared error (RMSE) was chosen as a way to emphasize large differences. Ordering the clusters by their average prediction error led to a simple way to identify outliers for the given dataset and model. Furthermore, as similar individual clusters contain similar configurations, prediction errors within clusters do not vary by a significant amount. For large datasets or models with computationally expensive predictions, this allows us to only predict a fraction of the cluster points rather than its entirety. In this work, we applied this to the most computationally expensive model of our selection (GAP/SOAP[28,29]), where only 1% of each cluster's points is predicted (for a minimum of 100 per cluster).

Poor predictions on specific clusters were commonly caused by the training set containing too few examples from the relevant region of CS. Often, this arose as a simple consequence of a non-optimal training set choice: out-of-equilibrium geometries are naturally rarer and thus less represented in datasets born from physical simulations. As such, a random choice of training points—even according to some statistical distribution—is very unlikely to contain those important out-of-equilibrium points. In other cases, poor predicted clusters contained configurations whose physiochemical properties deviate from the rest of the dataset. In such cases, even small changes in the geometry can lead to large differences in forces, hence the need to include a sizable contribution of outlying configurations to the training set for accurate predictions.

In the "improved model" step, all clusters with higher than average cluster prediction errors were recombined and reclustered more finely by applying the same agglomerative approach as before but with a larger number of clusters. This increased the resolution in which problematic regions of CS were identified, allowing for (a) filtering out of well-predicted configurations, previously buried in overly broad clusters, and (b) a finer distinction between all subregions of CS that include the configurations problematic for our initial model. Generally speaking, this step benefits from finer clusters, but those benefits quickly hit diminishing returns once the number of fine clusters exceeds the number of training points added at each step. As such, we chose both numbers to be equal for this work (100 points each).

From those fine clusters, we extracted their worst-predicted configurations to add to the training set. For this work, the number of training points added each step ("step size") is 100. Choosing the step size is largely a balance between increased training times (half the step size means roughly double the training time) and better final models. As always, decreasing the step size past a certain point provides negligible benefits, and as such, it was chosen to be 10% of the final model training set size in this work. For a given step size, the number of training points to extract from each individual fine cluster was proportional to the cluster's size as well as its prediction error. A weight was given to each fine cluster equal to the product of its total size and prediction error. Weights were then normalized such that the total weight across all clusters equaled the step size. Each cluster was given a number of points equal to the integer part of its weight; then, the remaining points were distributed one-by-one in the order of descending fractional part of the weights. Prioritizing larger clusters as well as outliers in terms of prediction accuracy meant that the method was able to fine-tune the training set earlier.

The complete training set for given data was created in an iterative manner, successively computing prediction errors, targeting problematic configurations to add to the training set, and retraining ML models. In the end, resulting models were trained on all the necessary data points to produce comparable prediction errors across all of CS within a dataset.

C. Alternatives

This subsection discusses some possible alternatives to the aforementioned approach and explains how they would affect the proposed method.

1. One could think of skipping the initial clustering and simply continue with the process using all configurations whose prediction error exceeds some minimum value. However, this would require calculating the prediction for every single data point, whereas our method allows calculating only a subset of each cluster as a representative error for the whole, saving the computational cost. For example, this is used for training the GAP/SOAP models in this work. This leads to two orders of magnitude reduced computational cost of the proposed method, saving hundreds of central processing unit (CPU) hours. Furthermore, the clusters and their prediction errors provide useful insight into the model's problematic regions of CS in a broad sense.

   In addition, many well-predicted clusters still contain singular configurations associated with high errors. As opposed to poorly predicted configurations inside poorly predicted clusters, the former are victims to the limitations of the ML model rather than those of the training set. Including those in the training set in an attempt to improve their prediction comes at a significant cost in accuracy in their otherwise well-predicted

cluster. Including entire clusters rather than singular points of high error anchors the aforementioned exceptions to their respective low-error cluster, thus allowing the algorithm to favor configurations that are representative of entire poorly predicted subregions of the CS.

2. One could also think of skipping the initial clustering step and immediately proceed to creating fine clusters from which we extract new training points. However, as previously explained, our clustering method of choice—agglomerative clustering—is not able to handle large datasets at once, and thus, a lower quality "approximation" of the method is used in the initial clustering step. The combination of all high-error clusters in the fine clustering step represent a subset much smaller than the original dataset, and thus, most, if not all, of the remaining data points can now be clustered in a single agglomerative step, leading to fine clusters of high quality.
3. Simple pairwise atomic distances were chosen as a descriptor over alternatives (e.g., Coulomb matrix) since our focus at the moment is still mostly on smaller molecules, where all atoms are important and can be handled at the same time without cutoffs. This choice, however, could be revised if needed when scaling to larger molecules, with dimensionality reduction techniques being of particular interest.
4. Finally, several methods were explored to choose which data points to extract from a fine cluster to the training set (for a given number of points). Selecting random points from the clusters already leads to improvements, but to a lesser extent than centroids (in the descriptor space) or points with the highest prediction error within their cluster. Both the latter methods performed similarly, however, as the outlier detection step already required the computation of prediction errors, the highest-prediction error criterion proved to be more efficient and is the default method for this paper.

IV. RESULTS AND DISCUSSION

The developed methodology was used to perform a detailed error analysis of three state-of-the-art MLFF models, namely, sGDML,[17] GAP[28] using the SOAP[29] descriptor, and SchNet.[18] For all models used, only the training set of the models was influenced: the training procedure itself is unaltered from the default. Unless otherwise specified, the settings for the models are the following: for SchNet, we use 128 features, a cutoff distance of 5 Å, four interaction blocks, 25 Gaussians, and a learning rate of 5×10^{-4}, and for GAP/SOAP, we use 12 radial and six angular functions and a cutoff of 5 Å (except for the uracil dataset, where eight radial and four angular functions were used for technical reasons). All sGDML models were trained on forces, SOAP/GAP models were trained on energies, and SchNet models were trained on both forces and energies.

The reference datasets used are of ethanol, salicylic acid, toluene, and uracil[39] for a total of 550k, 320k, 442k, and 133k number of points, respectively. In Sec. IV A, it will become clear that different ML models show varying performance on those molecular datasets; however, all of them are consistently inaccurate for out-of-equilibrium geometries (see Fig. 2). We show that by employing default training techniques, the prediction error on some physically relevant configurations can exceed the overall root mean squared error (RMSE) by up to a factor of 3. On the flip side, our improved training method alleviates the problem by creating models with significantly flattened errors across all configurations. This was applied to the previously mentioned datasets and an alanine tetrapeptide dataset generated in this work. In all cases, the ML models trained on the optimized training set are found to be significantly more reliable and stable for practical applications.

A. Outlier detection

In this subsection, we present the results of our outlier detection methods on uracil, salicylic acid, toluene, and ethanol molecules. The MLFF models of choice were sGDML, SchNet, and GAP/SOAP: we applied each model to the same molecular datasets. We cluster the datasets into 50 different regions of CS and compute the force prediction RMSE for every cluster. The results are plotted in Fig. 2. A very large disparity between the errors in clusters and the mean squared error can be observed, with some clusters presenting an error three times higher than the mean. The difference between them and the cluster of lowest error is of course even higher. It is clear that in these cases, a single MSE is not an appropriate metric to fully quantify how good the ML model works for the entire reference dataset. This is in direct contradiction with the idea of the MLFF being comparable to the underlying *ab initio* method, as entire regions of CS present an accuracy significantly worse than that of the reference calculations.

It is worth noting that each cluster corresponds to a different set of configurations. Hence, the proposed scheme detects poorly predicted regions of CS for a given model. An example geometry from the worst-predicted cluster is shown for salicylic acid: this configuration has a clear fingerprint of shared hydrogen between the carboxylic and hydroxyl groups. This process is a rare event in the reference database obtained by employing classical MD simulations and can be easily missed by the visualization of the trajectory or other human analyses. In contrast, the proposed clustering approach can easily separate such nontrivial configurations (a few hundred) from the overwhelming number (above 300k) of simple fluctuations around the equilibrium geometry. For the most part, however, clusters simply represent a subset of points whose geometry differs from the common (equilibrium) structures in various hard-to-interpret ways, which lead to them being under-represented in the dataset.

There are two possible reasons for the large prediction differences between clusters. First, poorly predicted regions can contain large fluctuations of molecular geometries, which are not well represented in the training set. This lack of information then hinders the ML models from learning those particular regions.

The second reason is non-trivial and can have significant impact on the reliability of the MLFF. The poorly predicted areas of the CS can represent physics or chemistry missing in the majority of the configurations in the reference dataset. This is showcased in our previous example of salicylic acid, where the cluster with the most significant error corresponds to a "shared" proton between the carboxylic and hydroxyl group. An accurate simulation of this process would require proper account of nuclear quantum effects, and hence, the corresponding configurations are a negligible minority

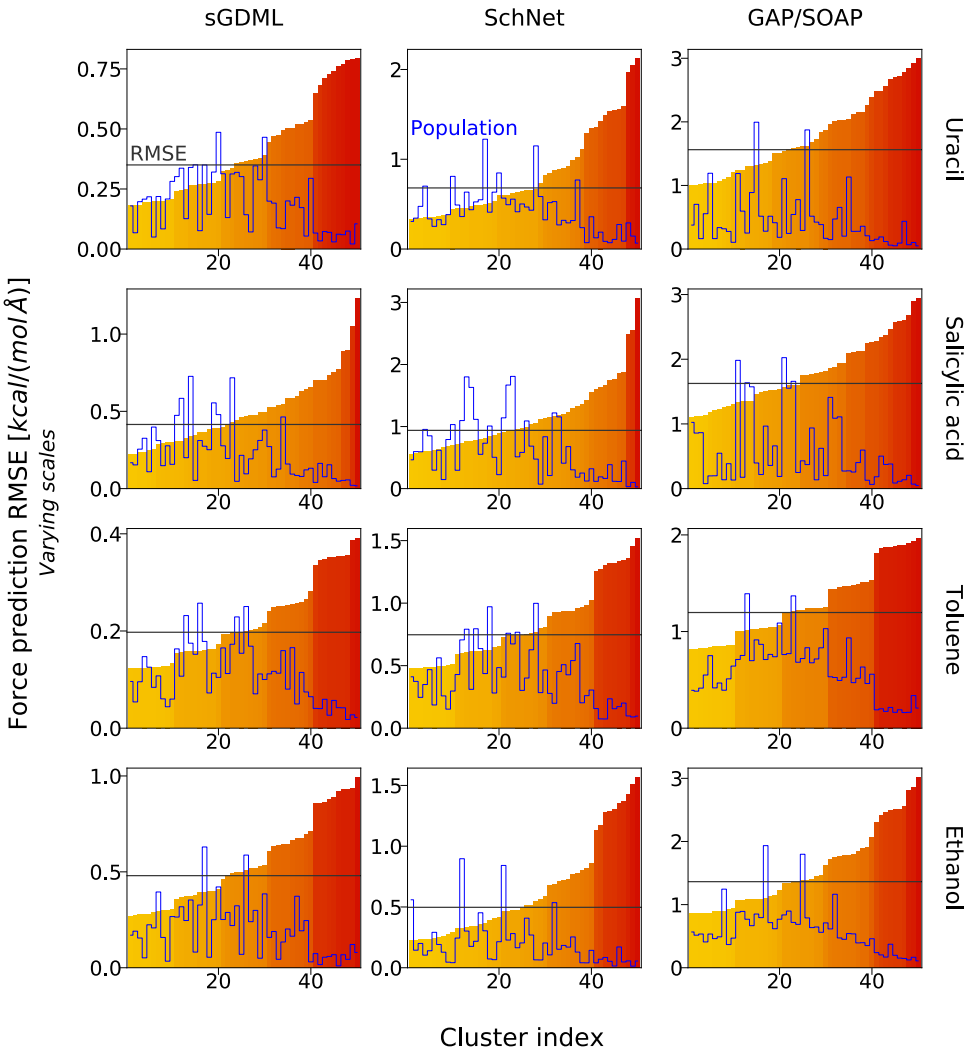

FIG. 2. Force prediction RMSE for sGDML, SchNet, and GAP/SOAP models on the ethanol, uracil, toluene, and salicylic acid datasets (y-axis, scale adapted for each model for better visibility) split into 50 clusters of similar configurations (x-axis) ordered by ascending error. RMSE (bars) is given on a per-cluster basis in contrast to the RMSE over the entire dataset (horizontal black solid line). Relative cluster populations are also indicated (blue solid line, arbitrary units).

(a few hundred) in the salicylic acid dataset containing over 320k molecular geometries of a classical MD run. Even if corresponding reference data were added to the MD dataset, those would be mainly ignored within the standard training schemes due to their relatively high energies. All in all, this points to the default MLFF being inapplicable for studying the proton sharing effect for our given dataset. In contrast, the developed method is designed to alleviate this problem by widening the model's applicability range to the fullest capability of the dataset. This is further expanded in Subsection IV D.

B. Optimizing the training set

We applied the improved training techniques developed in this work using sGDML, SchNet, and GAP/SOAP as our MLFF models. First, we performed the outlier detection by computing the root mean squared force prediction error on 50 clusters for an initial model with 200 training points. Note that for the GAP/SOAP model, only 1% of points of each cluster (minimum of 100) contributed to the error calculations. After that, 100 training points were added at every step for a total of eight steps, resulting in models of a total of 1000 training points each. All the details of the improved training procedure can be found in Sec. III. Specifically, the number of initial clusters (50) was empirically found to be appropriate for the datasets in this work: it is high enough to not mix largely different configurations while still avoiding the creation of clusters with extremely small population. The number of fine clusters was set to be equal to the step size (100), as benefits past this point show diminishing returns.

The main results of the developed improved training technique are shown in Fig. 3, representing the quality of MD simulations performed with the default and the improved models with respect to the reference method. Since forces are the variables entering the equations of motion, their errors are directly related to the deviations between the reference and ML trajectories (more so than the energies). While properties defined by the most common

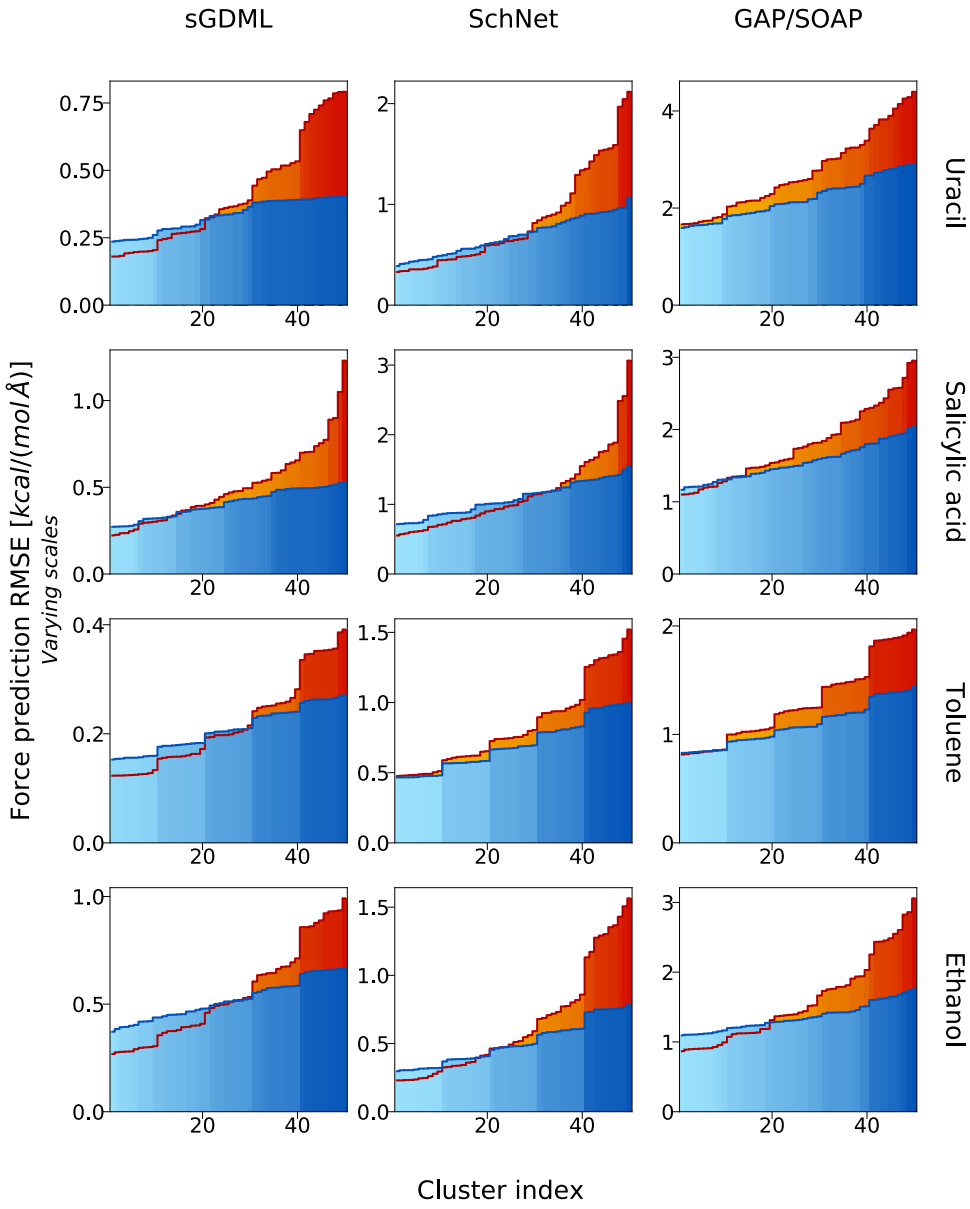

FIG. 3. Force prediction RMSE for sGDML, SchNet, and GAP/SOAP default models compared to the improved models (orange/blue bars, y-axis scale adapted for each model for better visibility). RMSE is computed on a per-cluster basis on ethanol, uracil, and salicylic acid datasets, split into 50 clusters of similar configurations (x-axis) ordered by ascending error.

configurations in the reference dataset—such as average energies at reasonably low temperatures—would be well represented by the default models, processes involving broad parts of the PES or regions under-represented in the reference dataset would be much better described using the proposed improved ML models.

The gradual increase in prediction accuracy throughout the iterative learning process can be seen on the example of salicylic acid in Fig. 4. The improved models are put side-to-side with default models of equal training set sizes for comparison.

It is important to note that the goal of the improved models is to present a more stable prediction error across all of the configurational space. They do so by explicitly including more out-of-equilibrium/rare configurations in their training set at the expense of the more common/in-equilibrium configurations in the dataset. As such, improvements in overall RMSE—which are mainly determined by accuracy on common configurations—are not the aim of this work. Nevertheless, while the overall RMSE on forces across the entire dataset does not change significantly, it sees some decrease for many of the molecules shown (see Table I).

C. Application to alanine tetrapeptide

So far, we applied the developed methods only on rather small molecules, demonstrating significant improvements in the

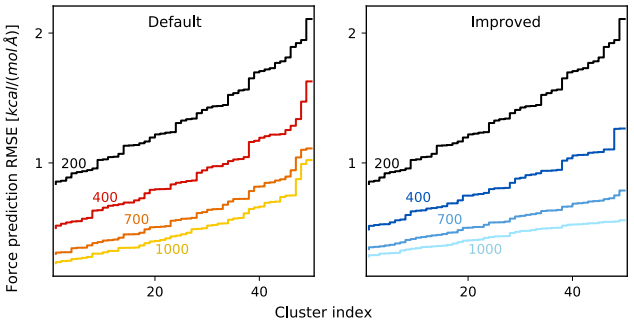

FIG. 4. Force prediction root mean squared error (solid lines) on all 50 clusters (x-axis) of the salicylic acid dataset ordered by ascending error. Different colors correspond to varying sizes of the training set, using the default sGDML training method (left) and the improved method (right).

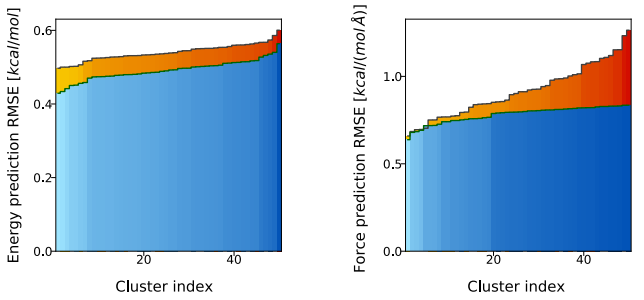

FIG. 5. Energy (left) and force (right) prediction RMSE for AcAla3NHMe SchNet default models on different clusters: default model (orange bars) compared to the improved model (blue bars). Each model consists of 6000 training points with identical training procedures and architecture.

TABLE I. Overall force RMSE for sGDML, SchNet, and GAP/SOAP models comparing default and improved versions. All numbers are given in $kcal/(mol\,\text{Å})$.

Molecule	Def. sGDML	Imp. sGDML	Def. SchNet	Imp. SchNet	Def. GAP	Imp. GAP
Uracil	0.38	0.32	0.77	0.65	2.71	2.17
Salicylic acid	0.44	0.39	0.99	1.03	1.80	1.54
Toluene	0.21	0.20	0.78	0.67	1.32	1.09
Ethanol	0.51	0.50	0.57	0.47	1.62	1.35

resulting MLFFs. In this subsection, we extend the applications to a noticeably larger molecule using an alanine tetrapeptide dataset. This peptide is large enough to exhibit several incipient secondary structure motifs akin to biological peptides and proteins. Our reference dataset was constructed via *ab initio* molecular dynamics at 500 K with the FHI-aims software[40] wrapped with the i-PI package[41] using the Perdew–Burke–Ernzerhof (PBE) exchange-correlation functional[42] with tight settings and the Many-Body Dispersion (MBD) method[43,44] to account for van der Waals interactions. The time step was set to 1 fs, and a global Langevin thermostat was used with a friction coefficient of 2 fs. In total, the dataset contains over 80k data points and covers at least three energy minima.

It is important to note that, especially for larger molecules, our training method can only lead to improvements if the base model has acceptable accuracy in the first place. Thus, we require a higher initial (also final) number of training points compared to previous datasets, as, here, we are trying to learn the PES of a much higher-dimensional and significantly more complex molecule. For the same reason, we use SchNet as our model in this subsection, since SchNet can be easily employed with much larger datasets than kernel-based methods (sGDML or GAP). For the improved model in this section, the initial training set size was set to 2000 points, with a step size of 500 for a final model of 6000 training points after eight total steps. The number of fine clusters was adjusted to 150 points to make up for the increased step size.

Figure 5 shows the performance of two equal-size well-converged SchNet models trained using the default and improved training schemes for identical SchNet architectures. Significant improvement can be found when applying our training methods to alanine tetrapeptide: both energy and force prediction errors are reduced for almost every cluster. Importantly, achieving the same improvement within the default training scheme would require adding more reference data to the training set. To highlight this, Fig. 6 shows the energy and force prediction accuracy of the SchNet models trained with different sizes of the training set.

Our method concentrates only on the improvement of the force predictions, where the RMSE of the default model for the worst clusters is about twice as large as that for the best one. The minor improvement in energy demonstrated in Fig. 5 is an accompanying effect, which we were not aiming for. RMSE for forces and energy drops only from 0.89 to 0.80 kcal/(mol Å) and 0.54 to 0.49 kcal/mol, respectively (Table II), but the flattening of the errors for the force prediction can have significant results in practice. See Subsection IV D for more details.

Improving the force predictions along with learning the energy within the SchNet model (or any other ML model) requires the employment of mixed loss functions, where the errors for energy and forces are minimized together. While this has proved to be a

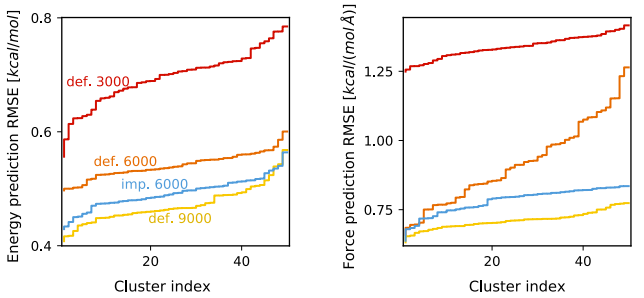

FIG. 6. Energy (left) and force (right) prediction RMSE for AcAla3NHMe SchNet default (orange) and improved (blue) models on different clusters. Comparing different sizes of training sets: 3000, 6000, and 9000 for default and 6000 for improved.

TABLE II. Overall force/energy RMSE for SchNet models of different training set sizes, comparing the default and improved versions.

RMSE	Default 3000	Default 6000	Improved 6000	Default 9000
Forces $[kcal/(mol\,Å)]$	1.34	0.89	0.79	0.71
Energy (kcal/mol)	0.70	0.54	0.48	0.46

successful method for various applications,[45–48] it is not the optimal choice when trying to create the best model for a given dataset. There, mixed loss functions are less efficient since optimizing two competing functions leads to sub-optimal results for each component, as both the energy and the force parts of the loss function are minimized by a different set of model parameters.[49] Our proposed method allows us to further improve energy or force predictions independently, merely by manipulating the training set.

In Fig. 5, we see that the cluster force predictions of the default model do not present a variance quite as large as the previously explored molecules. The main reason is that high-dimensional space (AcAla3NHMe has 42 atoms, i.e., 861 pairwise atomic distances) makes our clustering algorithms significantly weaker. Any distance metric loses meaning as the dimensionality increases, and clustering algorithms rely on the latter to subdivide datasets. As a consequence, our clusters are less well-defined and contain overlaps between qualitatively different configurations. This reduces the resolution between well and poorly predicted parts of CS, decreasing the efficiency of the proposed method. We expect that reducing the size of descriptors by making use of dimensionality reduction techniques (such as kernel principal component analysis) would improve the efficiency of the clustering schemes and, in turn, make the developed approach reliable for systems containing hundreds and thousands of atoms. A systematic extension of the developed approach to increasingly large systems will be the target of our future research and is beyond the scope of the current work.

In Fig. 6, one can observe three qualitatively different behaviors of the resulting SchNet models depending on the size of the training set: (a) Whenever the training set contains insufficient data (the model with 3k points), the constructed MLFF demonstrates low accuracy across the entire CS for both energy and forces. In this limit, the force-based improved training method proposed in this work does little to improve the FF since the starting model cannot distinguish between poorly and well-predicted areas of the CS. Without a robust initial model to base the iterations steps where training points are added, our methods fail to significantly improve the performance of an MLFF. (b) The training set contains enough data for the ML model to accurately learn the PES, but the forces are poorly predicted across CS (the model with 6k points), akin to previous examples (see Subsection IV B). This is precisely the scenario for which the proposed improved training technique has been developed. By comparing the default and improved models with 6k training points, one can see a significant boost in accuracy for forces accompanied by a slight improvement of the PES reconstruction (here, an improved model of 6k points is comparable to a default model of 7.5–8k points). As such, the proposed training method gives an optimal compromise between data efficiency and accuracy of ML models. (c) Finally, a training set overloaded with reference data (the model with 9k points) leaves little room for improvement. Indeed in this case, the training set contains all relevant configurations in the dataset (also, by extension, the validation set) such that the choice of training points becomes insignificant.

D. Practical benefits

1. MD of alanine tetrapeptide

To compare the performance of the improved and default peptide models, we ran constant-temperature MD simulations at 300 K and 400 K using the SchNet FF model. A Langevin thermostat with a friction coefficient of 100 fs was used and the time step was set to 0.5 fs to accurately reproduce fast hydrogen fluctuations in the molecule. Due to the size and high flexibility of the peptide, obtaining well-converged average energies requires MD trajectories of almost 2×10^6 steps, equivalent to 0.9 ns. Simulations of this size come at prohibitively expensive computational costs for any accurate *ab initio* method; MLFFs are the only way to perform them in practice. Note that our improved training procedure does come with higher computational costs (due to training the model multiple times), but the time spent on training is still very low compared to that of actually running the MD.

At 300 K, both models converge without any issues with a difference in average total energies of only 0.5 kcal/mol. The latter is within the accuracy of the ML models (see Fig. 6), meaning that both simulations give identical results. This is exactly what should be expected for a well-trained ML model in its zone of comfort. At 400 K, the situation changes drastically: the average total energy as a function of simulation time is shown in Fig. 7.

One can see that the default 6k model fails to reproduce the dynamics of the molecule at 400 K (as a zero energy level, we use the lowest potential energy in the reference dataset). The growth followed by an abrupt monotonic decay of the red curve advocates for the unreliability of the default training scheme for high-temperature simulations. As a result of wrong predictions, the molecule escapes

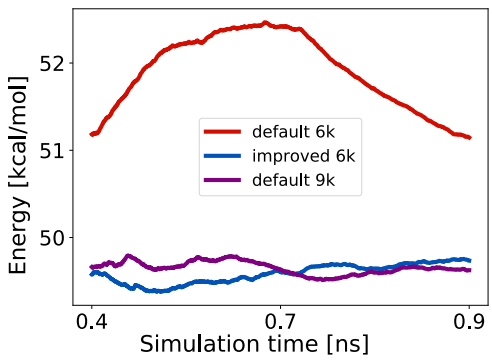

FIG. 7. Average total energy as a function of simulation time of the AcAla3NHMe molecule for the default/improved SchNet models with 6000 training points (blue/red) and a default SchNet model with 9000 training points (purple) for comparison. The constant-temperature MD simulations have been done at 400 K with 0.5 fs time step.

the applicability range of the MLFF and we observe nonphysical results. On the flip side, the improved 6k model remains stable all throughout the simulation, mimicking the results of the default 9k model despite its smaller training set size.

Note that the training set of all models was generated at 500 K and thus contain all the information needed for 400 K MD simulations. Importantly, increasing the temperature to generate new reference data would require broader sampling of parts of CS with computationally expensive *ab initio* methods—an unacceptable scenario for growing molecule sizes. Hence, the developed improved training scheme not only leads to quantitatively better predictions but also qualitatively increases the applicability range of the ML models by boosting their reliability.

2. Proton exchange in salicylic acid

To illustrate the usefulness of the improved models in out-of-equilibrium regions of CS, we created an artificial dataset of salicylic acid to act as a benchmark. The dataset was created by manually moving the hydrogen belonging to alcohol group between its bonded oxygen and the oxygen of the carboxylic group. The energy of the new configurations was computed using FHI-aims with equivalent settings to those of the original salicylic acid dataset.[39] The energy was then verified using the default and improved sGDML and SchNet models introduced in Sec. IV B, i.e., models trained on the original dataset with a training set size of 1000 points.

Figure 8 shows the predicted energies for the various models compared to the reference values as computed by FHI-aims. Note that for the most part, all values past 0.45 can be considered inside the extrapolation region of the models, as very few if any similar data points can be found inside the dataset on which the models were trained.

The default SchNet model very quickly deviates from the correct behavior by showing a shallow second minimum, followed by a weak energy barrier and finally a very unphysical drop in energy as the hydrogen atom gets closer to the hydroxyl group. The improved SchNet model completely avoids the second minimum and provides a stronger energy barrier before starting the unphysical drop in energy. The default sGDML model also shows a second minimum, which the improved sGDML model once again completely (and correctly) avoids, while also more closely matching the reference energy curve. Hence, in this extrapolation region, our improved training set selection procedure provides models whose behavior more qualitatively matches that of the reference method, which could have noticeable consequences in real applications. In particular, both default SchNet and sGDML models claim the existence of a proton sharing process between the alcohol and carboxylic groups. This would result in qualitatively wrong predictions when performing imaginary time path-integral MD simulations. In contrast, the improved training method proposed in this work removes this artifact on the PES, making such simulations reliable.

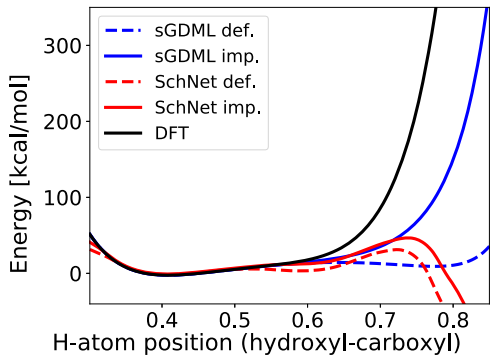

FIG. 8. Energy of salicylic acid (y-axis) for different positions of a shared hydrogen between the oxygen of the hydroxyl and carboxyl group (x-axis, 0 = on top of hydroxyl oxygen and 1 = on top of carboxyl oxygen). Several models are compared to the reference values (black solid curve): default sGDML (blue dashed curve), improved sGDML (blue solid curve), default SchNet (red dashed curve), and improved SchNet (red solid curve). Reference DFT values computed with FHI-aims with the equivalent settings to those of the original salicylic acid dataset.[39]

V. CONCLUSIONS

By leveraging supervised and unsupervised ML, we proposed a new strategy for improved training set selection for the construction of molecular machine learning force fields. We developed an automatic outlier detection method that exposed a noticeable bias in the predictive accuracy of the ML models toward common/in-equilibrium configurations at the expense of rarer/out-of-equilibrium ones, leading to entire regions of CS with significantly higher-than-average prediction errors. Our procedure is able to extract tiny subsets of molecular configurations representing nontrivial physical or chemical processes from an overwhelming amount of reference data. For example, a few hundred configurations with fingerprints of a shared proton in the salicylic acid molecule were found within more than 300k classical fluctuations around the equilibrium state.

The developed error analysis helped optimize the training set choice, resulting in the largely improved accuracy of ML models across all of CS—effectively "flattening" the prediction error curve throughout the input space. During the training process, we iteratively selected poorly predicted training points from different parts of CS to add to the training set. This ensured that it contained sufficient representation from every qualitatively different type of the configurations in the reference dataset. Models born from this approach proved to be more reliable than those with training sets in line with the dataset's inherent distributions and guarantee "chemical accuracy" for the entire sampled CS. Among other things, this enables making fair and unbiased comparisons between the expressive power of different ML models for reproducing PESs.

With the examples of small organic molecules and an alanine tetrapeptide, we demonstrated that the developed training method leads to an optimal compromise between the data efficiency and accuracy of MLFFs, avoiding the need to generate extensive amounts of computationally expensive highly accurate reference data for training sets. Along with quantitative reductions in prediction errors, the ML models trained on the optimized training sets offer qualitative improvements in reliability for practical applications. This is demonstrated in the example of high-temperature MD

simulations for alanine tetrapeptide and in a hydrogen exchange mechanism for salicylic acid. Future plans include combining the developed approach to dimensionality reduction techniques to extend the applicability range to systems consisting of hundreds and thousands of atoms.

While this paper focused on improving three specific ML models (GAP with SOAP descriptors and sGDML as kernel based approaches and SchNet as a neural network), all methods can be easily extended to any ML field presenting similar training problems. The code for the outlier detection and improved training is available in the open-source software MLFF on GitHub.[27]

SUPPLEMENTARY MATERIAL

See the supplementary material for the dataset introduced and used in Sec. IV C constructed via *ab initio* molecular dynamics at 500 K with the FHI-aims software[40] wrapped with the i-PI package[41] using the Perdew–Burke–Ernzerhof (PBE) exchange-correlation functional[42] with tight settings and the Many-Body Dispersion (MBD) method[43,44] to account for van der Waals interactions. The timestep was set to 1 fs and a global Langevin thermostat was used with a friction coefficient of 2 fs. In total, the dataset contains over 80k data points and covers at least three energy minima.

ACKNOWLEDGMENTS

We acknowledge financial support from the Luxembourg National Research (FNR) under the AFR Project (Grant Nos. 14593813 and FNR C19/MS/13718694/QML-FLEX), FNR DTU-PRIDE MASSENA, and the European Research Council (ERC-CoG Grant No. BeStMo).

DATA AVAILABILITY

The data that support the findings of this study are openly available on the sGDML[17,39] official website and the supplementary material.

REFERENCES

[1]S. A. Hollingsworth and R. O. Dror, "Molecular dynamics simulation for all," Neuron **99**, 1129–1143 (2018).
[2]J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. Kalé, and K. Schulten, "Scalable molecular dynamics with NAMD," J. Comput. Chem. **26**, 1781–1802 (2005).
[3]T. Hansson, C. Oostenbrink, and W. van Gunsteren, "Molecular dynamics simulations," Curr. Opin. Struct. Biol. **12**, 190–196 (2002).
[4]M. J. Abraham, T. Murtola, R. Schulz, S. Páll, J. C. Smith, B. Hess, and E. Lindahl, "GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers," SoftwareX **1-2**, 19–25 (2015).
[5]D. C. Rapaport, *The Art of Molecular Dynamics Simulation* (Cambridge University Press, 2004).
[6]N. Plattner and F. Noé, "Protein conformational plasticity and complex ligand-binding kinetics explored by atomistic simulations and Markov models," Nat. Commun. **6**, 7653 (2015).
[7]K. A. Krylova, J. A. Baimova, I. P. Lobzenko, and A. I. Rudskoy, "Crumpled graphene as a hydrogen storage media: Atomistic simulation," Physica B **583**, 412020 (2020).
[8]Y. Wu, H. Sun, L. Wu, and J. D. Deetz, "Extracting the mechanisms and kinetic models of complex reactions from atomistic simulation data," J. Comput. Chem. **40**, 1586–1592 (2019).
[9]S. Wolf, M. Amaral, M. Lowinski, F. Vallée, D. Musil, J. Güldenhaupt, M. K. Dreyer, J. Bomke, M. Frech, J. Schlitter, and K. Gerwert, "Estimation of protein-ligand unbinding kinetics using non-equilibrium targeted molecular dynamics simulations," J. Chem. Inf. Model. **59**, 5135–5147 (2019).
[10]D. T. Kallikragas and I. M. Svishchev, "Atomistic simulations of corrosion related species in nano-cracks," Corros. Sci. **135**, 255–262 (2018).
[11]H. DorMohammadi, Q. Pang, L. Árnadóttir, and O. Burkan Isgor, "Atomistic simulation of initial stages of iron corrosion in pure water using reactive molecular dynamics," Comput. Mater. Sci. **145**, 126–133 (2018).
[12]I. B. Obot, K. Haruna, and T. A. Saleh, "Atomistic simulation: A unique and powerful computational tool for corrosion inhibition research," Arabian J. Sci. Eng. **44**, 1–32 (2019).
[13]R. B. Best, "Atomistic molecular simulations of protein folding," Curr. Opin. Struct. Biol. **22**, 52–61 (2012).
[14]H. Xiao, B. Huang, G. Yao, W. Kang, S. Gong, H. Pan, Y. Cao, J. Wang, J. Zhang, and W. Wang, "Atomistic simulation of the coupled adsorption and unfolding of protein GB1 on the polystyrenes nanoparticle surface," Sci. China: Phys., Mech. Astron. **61**, 038711 (2018).
[15]D. Meneksedag-Erol and S. Rauscher, "Atomistic simulation tools to study protein self-aggregation," in *Protein Self-Assembly: Methods and Protocols*, Methods in Molecular Biology, Vol. 2039, (Springer, 2019), pp. 243–262.
[16]F. Noé, A. Tkatchenko, K.-R. Müller, and C. Clementi, "Machine learning for molecular simulation," Annu. Rev. Phys. Chem. **71**, 361–390 (2020).
[17]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).
[18]K. T. Schütt, P.-J. Kindermans, H. E. Sauceda, S. Chmiela, A. Tkatchenko, and K.-R. Müller, "SchNet: A continuous-filter convolutional neural network for modeling quantum interactions," in *Proceedings of the 31st International Conference on Neural Information Processing Systems* (2017), Vol. 30, pp. 992–1002.
[19]G. Montavon, M. Rupp, V. Gobre, A. Vazquez-Mayagoitia, K. Hansen, A. Tkatchenko, K.-R. Müller, and O. Anatole von Lilienfeld, "Machine learning of molecular electronic properties in chemical compound space," New J. Phys. **15**, 095003 (2013).
[20]J. Behler, "Perspective: Machine learning potentials for atomistic simulations," J. Chem. Phys. **145**, 170901 (2016).
[21]K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko, "Quantum-chemical insights from deep tensor neural networks," Nat. Commun. **8**, 13890 (2017).
[22]K. Hansen, F. Biegler, R. Ramakrishnan, W. Pronobis, O. Anatole von Lilienfeld, K.-R. Müller, and A. Tkatchenko, "Machine learning predictions of molecular properties: Accurate many-body potentials and nonlocality in chemical space," J. Phys. Chem. Lett. **6**, 2326–2331 (2015).
[23]A. Mardt, L. Pasquali, H. Wu, and F. Noé, "VAMPnets for deep learning of molecular kinetics," Nat. Commun. **9**, 5 (2018).
[24]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).
[25]F. A. Faber, A. Lindmaa, O. A. von Lilienfeld, and R. Armiento, "Machine learning energies of 2 million elpasolite (ABC_2D_6) crystals," Phys. Rev. Lett. **117**, 135502 (2016).
[26]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).
[27]F. Gregory, MLFF, https://github.com/fonsecag/MLFF (2020).
[28]A. P. Bartók and G. Csányi, "Gaussian approximation potentials: A brief tutorial introduction," Int. J. Quantum Chem. **115**, 1051–1057 (2015).
[29]A. P. Bartók, R. Kondor, and G. Csányi, "On representing chemical environments," Phys. Rev. B **87**, 184115 (2013).
[30]Q. Lin, Y. Zhang, B. Zhao, and B. Jiang, "Automatically growing global reactive neural network potential energy surfaces: A trajectory-free active learning strategy," J. Chem. Phys. **152**, 154104 (2020).

[31] J. S. Smith, B. Nebgen, N. Lubbers, O. Isayev, and A. E. Roitberg, "Less is more: Sampling chemical space with active learning," J. Chem. Phys. **148**, 241733 (2018).
[32] V. Botu and R. Ramprasad, "Adaptive machine learning framework to accelerate *ab initio* molecular dynamics," Int. J. Quantum Chem. **115**, 1074–1083 (2015).
[33] M. Ceriotti, G. A. Tribello, and M. Parrinello, "Demonstrating the transferability and the descriptive power of sketch-map," J. Chem. Theory Comput. **9**, 1521–1532 (2013).
[34] 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).
[35] J. H. Ward, Jr., "Hierarchical grouping to optimize an objective function," J. Am. Stat. Assoc. **58**, 236–244 (1963).
[36] D. Sculley, "Web-scale k-means clustering," in *Proceedings of the 19th International Conference on World Wide Web, WWW '10* (Association for Computing Machinery, New York, NY, 2010), pp. 1177–1178.
[37] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (SODA, 2007), Vol. 8, pp. 1027–1035.
[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res. **12**, 2825–2830 (2011).
[39] 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).
[40] V. Blum, R. Gehrke, F. Hanke, P. Havu, V. Havu, X. Ren, K. Reuter, and M. Scheffler, "*Ab initio* molecular simulations with numeric atom-centered orbitals," Comput. Phys. Commun. **180**, 2175–2196 (2009).
[41] V. Kapil, M. Rossi, O. Marsalek, R. Petraglia, Y. Litman, T. Spura, B. Cheng, A. Cuzzocrea, R. H. Meißner, D. M. Wilkins, B. A. Helfrecht, P. Juda, S. P. Bienvenue, W. Fang, J. Kessler, I. Poltavsky, S. Vandenbrande, J. Wieme, C. Corminboeuf, T. D. Kühne, D. E. Manolopoulos, T. E. Markland, J. O. Richardson, A. Tkatchenko, G. A. Tribello, V. Van Speybroeck, and M. Ceriotti, "i-PI 2.0: A universal force engine for advanced molecular simulations," Comput. Phys. Commun. **236**, 214–223 (2019).
[42] J. P. Perdew, K. Burke, and M. Ernzerhof, "Generalized gradient approximation made simple," Phys. Rev. Lett. **77**, 3865–3868 (1996).
[43] A. Ambrosetti, A. M. Reilly, R. A. DiStasio, and A. Tkatchenko, "Long-range correlation energy calculated from coupled atomic response functions," J. Chem. Phys. **140**, 18A508 (2014).
[44] A. Tkatchenko, R. A. DiStasio, R. Car, and M. Scheffler, "Accurate and efficient method for many-body van der Waals interactions," Phys. Rev. Lett. **108**, 236402 (2012).
[45] J. Westermayr and P. Marquetand, "Deep learning for UV absorption spectra with SchNarc: First steps toward transferability in chemical compound space," J. Chem. Phys. **153**, 154112 (2020).
[46] J. Westermayr, F. A. Faber, A. S. Christensen, O. A. von Lilienfeld, and P. Marquetand, "Neural networks and kernel ridge regression for excited states dynamics of $CH_2NH^+_2$: From single-state to multi-state representations and multi-property machine learning models," Mach. Learn.: Sci. Technol. **1**, 025009 (2020).
[47] F. A. Faber, A. S. Christensen, and O. A. von Lilienfeld, "Quantum machine learning with response operators in chemical compound space," in *Machine Learning Meets Quantum Physics*, edited by K. T. Schütt, S. Chmiela, O. A. von Lilienfeld, A. Tkatchenko, K. Tsuda, and K.-R. Müller (Springer International Publishing, Cham, 2020), pp. 155–169.
[48] A. S. Christensen and O. A. von Lilienfeld, "On the role of gradients for machine learning of molecular energies and forces," Mach. Learn.: Sci. Technol. **1**, 045018 (2020).
[49] O. T. Unke, S. Chmiela, H. E. Sauceda, M. Gastegger, I. Poltavsky, K. T. Schütt, A. Tkatchenko, and K.-R. Müller, "Machine learning force fields," Chem. Rev. (published online, 2021).