RESEARCH ARTICLE | JANUARY 02 2025

Active learning of molecular data for task-specific objectives

Kunal Ghosh ; Milica Todorović ; Aki Vehtari ; Patrick Rinke

J. Chem. Phys. 162, 014103 (2025)
https://doi.org/10.1063/5.0229834

The Journal of Chemical Physics ARTICLE pubs.aip.org/aip/jcp

Active learning of molecular data for task-specific objectives

Cite as: J. Chem. Phys. 162, 014103 (2025); doi: 10.1063/5.0229834
Submitted: 19 July 2024 • Accepted: 12 December 2024 •
Published Online: 2 January 2025

Kunal Ghosh,[1,2] Milica Todorović,[3] Aki Vehtari,[2] and Patrick Rinke[1,4,5,6,a)]

AFFILIATIONS

[1] Department of Applied Physics, Aalto University, P.O. Box 11000, FI-00076 Aalto, Finland
[2] Department of Computer Science, Aalto University, P.O. Box 15400, FI-00076 Aalto, Finland
[3] Department of Mechanical and Materials Engineering, University of Turku, FI-20014 Turku, Finland
[4] Physics Department, TUM School of Natural Sciences, Technical University of Munich, Garching, Germany
[5] Atomistic Modelling Center, Munich Data Science Institute, Technical University of Munich, Garching, Germany
[6] Munich Center for Machine Learning (MCML), Munich, Germany

[a)] **Author to whom correspondence should be addressed:** patrick.rinke@tum.de

ABSTRACT

Active learning (AL) has shown promise to be a particularly data-efficient machine learning approach. Yet, its performance depends on the application, and it is not clear when AL practitioners can expect computational savings. Here, we carry out a systematic AL performance assessment for three diverse molecular datasets and two common scientific tasks: compiling compact, informative datasets and targeted molecular searches. We implemented AL with Gaussian processes (GP) and used the many-body tensor as molecular representation. For the first task, we tested different data acquisition strategies, batch sizes, and GP noise settings. AL was insensitive to the acquisition batch size, and we observed the best AL performance for the acquisition strategy that combines uncertainty reduction with clustering to promote diversity. However, for optimal GP noise settings, AL did not outperform the randomized selection of data points. Conversely, for targeted searches, AL outperformed random sampling and achieved data savings of up to 64%. Our analysis provides insight into this task-specific performance difference in terms of target distributions and data collection strategies. We established that the performance of AL depends on the relative distribution of the target molecules in comparison to the total dataset distribution, with the largest computational savings achieved when their overlap is minimal.

I. INTRODUCTION

In recent years, applications of machine learning (ML) in material science have produced a plethora of new discoveries,[1–3] such as the discoveries of millions of novel molecules,[4] phase-change materials,[5] and metallic glasses.[6] These discoveries rely on accurate property predictions by ML models, which usually require large amounts of training data.[7–9] Such large training datasets are costly to compile for supervised ML tasks[10–13] because the property labels are obtained from expensive simulations or time-consuming experiments.

Materials datasets are mostly compiled by human experts,[14] which can lead to bias and redundancy.[15] Instead of collecting larger training datasets to mitigate biases, we propose the use of active learning[16] (AL) with pool-based sampling to compile smaller datasets that are free from human bias. AL iteratively improves the performance of ML models by intelligently compiling the model's training data via acquisition functions. The strategic data-driven compilation reduces dataset redundancies in comparison to human compilation. Moreover, different acquisition strategies (AS) can be exploited to produce datasets specifically designed for targeted ML tasks (see Fig. 1).

In materials science, AL has been deployed in the development of novel algorithms[17–20] and applications for material discovery and property prediction tasks.[21–25] While AL holds considerable promise for smart data collection,[26–28] just as often failure or no improvements are reported (some of these reports are anecdotal since failures are rarely published).[29–31] To address the apparent

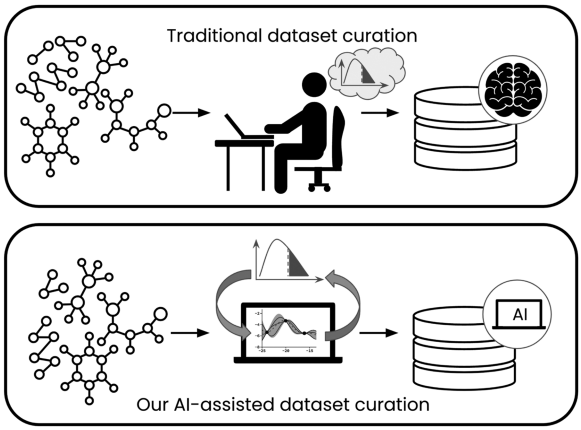

FIG. 1. Traditionally, materials datasets with a target property are compiled combinatorially, using human intuition. We propose an AI-assisted dataset compilation scheme that learns to identify materials with the target property iteratively. The identification improves with each iteration and is free from human bias.

discrepancy, we present a systematic study into the performance and benefits of AL for the compilation of molecular datasets. We compare the benefits of AL based on Gaussian process regression (GPR) in dataset compilation for two different ML tasks: dataset pruning and inverse material design.

The objective of the first task is to generate a maximally compact and informative dataset (task 1). In this very common AL use case, one seeks to balance the redundancy and diversity of materials data. We propose batch acquisition[32,33] strategies (AS) for AL and assess their performance for a variety of parameter choices and different materials datasets. We seek to identify the acquisition strategies that generate the smallest training sets and lead to the best model performance.

In the second dataset compilation task, we address the targeted materials property search (task 2). This objective emulates the ML use case of backward prediction: given a property target, we seek the best candidate materials with this feature. For the ML model to accurately predict the feature, the AL-compiled dataset should balance data entries with and without this information. We use the AS approaches from task 1 to explore different routes toward dataset assembly for this task. Our objective here is to identify the approach that maximizes the predictive accuracy of the ML model on task 2.

This study employs three molecular datasets with different levels of complexity and redundancy.[15] We focus on learning the ionization potential, equivalent to the energy of the highest occupied molecular orbital (HOMO) computed by *ab initio* simulations. In task 2, we seek to identify molecules with a target property: HOMO values greater than ε. We make use of pre-labeled datasets to accelerate our study: starting from the large pool of possible molecular structures, we draw data points with AL and include labels in the ML model. This is analogous to a realistic AL use case, where researchers might start with a large pool of unlabeled materials structures and perform computations to label them as needed.[28] Smaller training datasets would require fewer calculations to obtain the material property labels for ML training. Therefore, the outcome of task 1 is to demonstrate whether computational savings can be achieved by generating maximally informative, compact datasets. The outcome of task 2 is to compile a list of materials that match a targeted property and, consequently, identify optimal materials for a technological application.

II. DATASETS

A. AA (44 k amino acids and dipeptides)

The amino acid (AA) dataset[34] contains 44 004 isolated and cation-coordinated conformers of 20 proteinogenic amino acids and their amino-methylated and acetylated (capped) dipeptides. The molecules reach up to 39 atoms in size and include the chemical elements H, C, N, O, and S, as well as divalent cations (Ca^{2+}, Ba^{2+}, Sr^{2+}, Cd^{2+}, Pb^{2+}, and Hg^{2+}). The amino acid conformers encode different protonation states of the backbone and the side chains. Since all amino acids share a common backbone, the complexity of this dataset lies in differing side chains, dihedral angles, and metal cations. The AA dataset was generated by conformational sampling, and all molecular structures and properties were calculated with density-functional theory (DFT) using the Perdew–Burke–Ernzerhof (PBE)[35] exchange–correlation functional with Tkatchenko–Scheffler van der Waals corrections (vdW).[36] AA was used to benchmark several ML models[3,15,37,38] and clustering techniques.[39] The HOMO energy values of the AA dataset follow a bimodal distribution, with one mode approximately at −14 eV and the second mode at −5 eV. The distribution has its mean at −10.85 eV, with a standard deviation of 3.97 eV, and ranges between −19.63 and −0.06 eV.

B. QM9 (134 k organic molecules)

The QM9 dataset[14] is a subset of the GDB-17 database,[40] which was compiled by enumerating all organic molecules that contain up to 17 atoms of C, N, O, S, and halogen elements. The QM9 dataset features the first 133 814 molecules from the GDB-17 database. It contains small organic molecules with up to 9 heavy atoms (C, N, O, and F), which comprise 621 stoichiometries of small amino acids and nucleobases (pharmaceutically relevant organic building blocks). Molecular structures and labels were computed at the PBE + vdW DFT level by Stuke *et al.*[15] Despite considerable redundancy, the QM9 dataset was used in a variety of ML studies and has become the drosophila of ML in chemistry. The HOMO energy values of the QM9 dataset follow a unimodal distribution, with the mode approximately at −5.78 eV. The distribution has its mean at −5.77 eV, with a standard deviation of 0.52 eV, and ranges between −10.45 and −2.53 eV.

C. OE (64 k opto-electronically active molecules)

The opto-electronic (OE) dataset[41] consists of 64 710 large organic molecules with up to 174 atoms. The structures were extracted from monomolecular organic crystals in the Cambridge Structural Database[42] (CSD) by Schober *et al.*[43,44] for their high charge carrier mobility and re-optimized in vacuum at the PBE + vdW level of DFT theory. OE contains molecules with 16 different elements: H, Li, B, C, N, O, F, Si, P, S, Cl, As, Se, Br, Te, and I. The molecular structures are more complex than in QM9 and AA, with large conjugated backbones and unusual functional

groups. Of the three datasets in this work, OE offers the largest chemical diversity, both in terms of molecule size and the number of different elements. OE has become one of the benchmark datasets for molecule generation[13] and property prediction.[10,45] The HOMO energy values of the OE dataset follow a skewed unimodal distribution, with the mode approximately at −7.56 eV. The distribution has its mean at −5.49 eV, with a standard deviation of 0.57 eV, and ranges between −12.67 and −2.73 eV.

III. METHODOLOGY

To establish the AL approach in this study, we first derived the AL workflow for guided dataset compilation and proposed several ASs for picking data. Next, we selected the materials descriptor, ML model, and the metrics for evaluating the success of AL. Since we have a fixed set of molecules to choose from, we utilize pool-based ASs, as compared to query synthesis and streaming selection strategies,[16] which are more suited for generative models and situations where data are obtained in a stream, sequentially. We describe all these steps and their implementation below.

AL molecular dataset compilation is the process of assembling the ML training set in an iterative way by selecting groups of data points from a large pool of molecules called the held-out set. As illustrated in Fig. 2(a), each batch would typically be labeled by DFT simulations before the GPR ML models[46] are fit to the training set. GPR outcomes are utilized in ASs that determine how best to select the next batch of data for maximum improvement of ML models, given the objective.

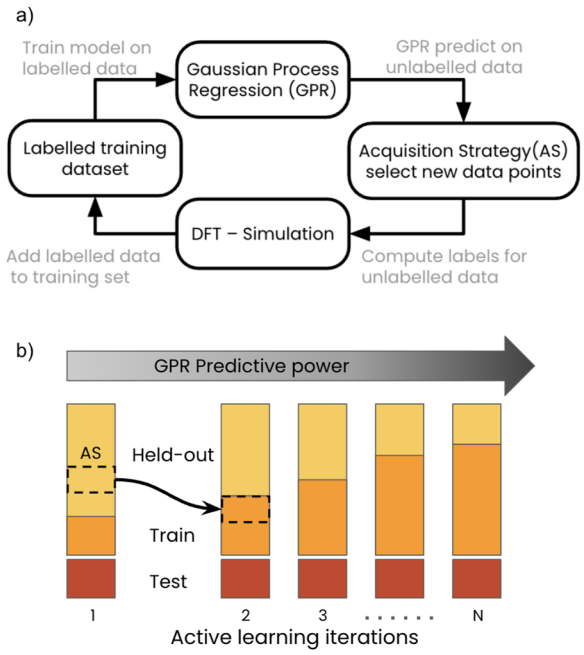

FIG. 2. Illustration of AL steps for (a) the active learning iteration and (b) the evolution of held-out, train, and test set sizes. Before performing active learning, small, labeled training and test sets are compiled. A Gaussian process regression (GPR) model is fitted to the training set and then used to obtain property predictions of the unlabeled held-out set. The acquisition strategy (AS) combines the predicted property, the corresponding prediction uncertainty, and molecular representation to select molecules from the held-out set. Selected molecules are then labeled using *ab initio* simulation software (DFT) and added to the training set. The larger training set is used to train a new GPR, and the iteration continues.

We encoded this procedure into our AL workflow, depicted in Fig. 2(b). Before the start, a batch of molecules was set aside from the held-out set to form the test set, which was kept fixed throughout all AL experiments. The test set served to evaluate the performance of the ML models with the evolving training set. At the start, the first batch of $\mathbf{N}_{\text{init}}$ molecules was selected from the held-out set to nucleate the training set and fit the first GPR model. The model was then applied to predict the property for all the remaining molecular structures in the held-out set. We utilized the prediction results to construct a selection criterion for the AS, typically referred to as the oracle in the AL literature.[16] Consequently, a batch of $\mathbf{N}_b$ molecules was selected from the held-out set and appended to the training set. Here, DFT simulations would typically be required to label the molecular structures, but we expedited the tests with pre-computed labels. A fully labeled augmented training set was the outcome of a single AL iteration. The next AL iteration began by retraining the GPR with the updated training set.

The quality of the compiled dataset depends on how intelligently an AS can pick molecules from the held-out set. The AL literature is rife with various AS designs, for example, strategies for compiling thin-film materials,[47] potentials for metal-organics,[25] or the design of layered materials.[48] While these strategies addressed specific tasks (task 2 here), it is also important to consider AS designs for general ML accuracy with minimal datasets (task 1 here).

The selection strategy in acquisition functions is typically based on a trade-off between diversity and redundancy. Increasing diversity ensures a good representation of different kinds of molecules and reflects data space exploration. Higher redundancy allows models to learn minor variations in property values of similar molecules through data exploitation. In practice, the trade-off is implemented through considerations of Gaussian processes (GP) prediction uncertainty and clustering algorithms. When GP models are applied to the molecular structures in the held-out set, prediction uncertainty is the highest for the structures that differ most from the molecules in the training set. Uncertainty-based picking thus increases the diversity of the training set. However, molecules with high prediction uncertainty could all be similar to each other. Similarity among the selected structures can be minimized by clustering them and choosing data from each cluster. The number of clusters correlates with the diversity of the selection, and the number of molecules selected from each cluster determines the redundancy.

While all these considerations are relevant, it is unclear which combination of AS choices and related parameters would lead to the most accurate ML models, trained on the most compact, maximally informative datasets. To address this question, we constructed five AS and compared them in the AL workflows described above. The following AS selection rules are illustrated in Fig. 3).

A. Random

In the simplest strategy, we randomly selected a fixed number of molecules from the held-out set, labeled them, and added the

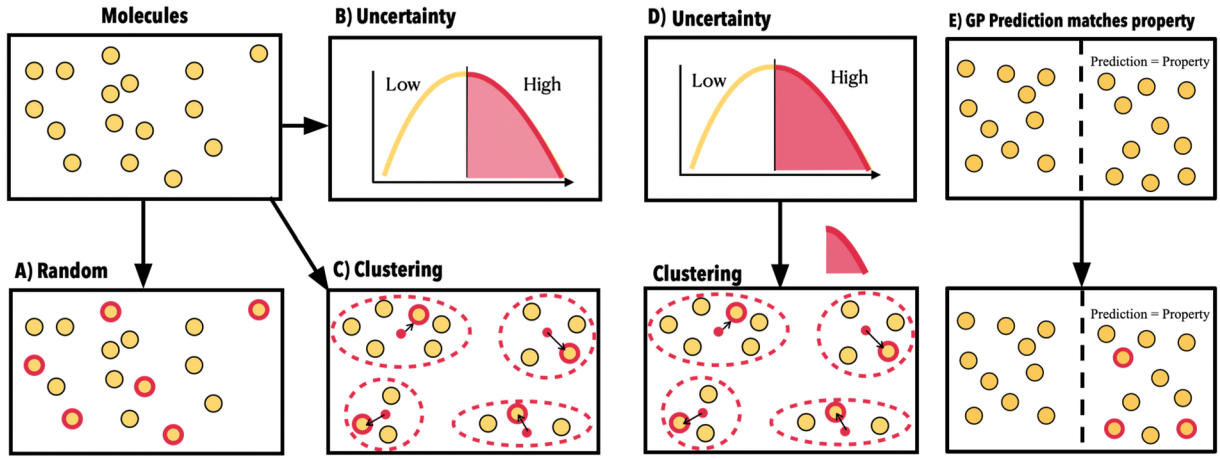

FIG. 3. Illustration of active learning acquisition strategies: (a) random; (b) utilizing GPR prediction uncertainty; (c) by clustering molecular representations; (d) by first selecting molecules with high GPR prediction uncertainty and then clustering the selected molecules, selecting the cluster centers; and (e) by selecting a set of molecules with GPR predicted property lying within a property value range. Subsequently, a random selection is made from the previous set. The round yellow circles indicate molecules. Round circles, with a thick red border, illustrate selected molecules. Dashed lines separate groups of molecules and red dashed lines indicate clusters of molecules. The red dot inside a cluster indicates the cluster center, and the arrow illustrates the molecule closest to the cluster center.

data to the training set. Such sampling ensures an even representation of data across the held-out set but also encodes all biases and redundancies. This traditional sampling strategy often yields accurate machine learning models.[49] It served as the baseline against which we compared AL approaches.

B. Uncertainty

This strategy utilized GP prediction uncertainty to add molecules to the training set. At each iteration, after training the GP, we computed the predictions and corresponding uncertainty values on the held-out set molecules. Data indices were sorted based on their prediction uncertainty, and a batch of $\mathbf{N}_b$ molecules with the highest uncertainty was selected. This AS encourages exploration and leads to rapid ML uncertainty reduction, without any considerations of diversity.

C. Clustering

Molecules in the held-out set were grouped into $\mathbf{N}_b$ clusters. From each group, one structure closest to the cluster center was selected and added to the training set. The objective here was to maximize the structural diversity of molecules in the training set, irrespective of uncertainty.

D. Uncertainty and clustering

The trained GP model was applied to compute predictions on the held-out set, after which the data were sorted by GP prediction uncertainty. The top 50% of the molecules with the highest uncertainty were selected and divided into $\mathbf{N}_b$ clusters. Molecules closest to cluster centers were added to the training set. This AS combines the two previous ones to overcome their respective shortcomings.

E. Property search

This AS was specifically designed for task 2. The trained GP model was applied to compute property predictions on the held-out set, after which the molecular structures were filtered based on the target property criterion (here, HOMO $> \varepsilon$). From all the molecules with this predicted property, $\mathbf{N}_b$ structures were chosen at random and added to the training set. In the early stages of AL, poorer GP model accuracy may lead to less accurate selections. As iterations proceed, more molecules matching the search criterion should be found, emulating data exploitation.

The key AS parameter is the batch size ($\mathbf{N}_b$), the number of molecules added to the training set with each iteration. $\mathbf{N}_b$ can be fixed (e.g., $\mathbf{N}_{\text{const}} = 1000$ or 1 k molecules) or adaptive, evolving with each iteration. Adaptive $\mathbf{N}_b$ could prove important because GP models may have limited accuracy in the early iterations of active learning, favoring small $\mathbf{N}_b$. However, proceeding with small batches would be inefficient later on, because frequent GPR fitting would slow down AL. During later stages, when model accuracy improves and molecular properties are predicted more reliably, larger batches are better suited. We implemented a power law (POW) batch scheme to evolve $\mathbf{N}_b$ with iteration t as $N_b(t) = 2^t * \mathbf{N}_{\text{const}}$, where $\mathbf{N}_{\text{const}} = 1000$. This meant that the batch size doubled in size with each AL iteration t, making the training set size $\mathbf{N}_{TR}(t)$,

$$\mathbf{N}_{TR}(t) = 2^t * \mathbf{N}_{\text{const}} + \mathbf{N}_{\text{init}} \qquad \forall\ t \in \mathbb{N}. \tag{1}$$

We also tested a constant batch scheme, where $\mathbf{N}_{\text{const}}$ was either 1, 2, 4, or 8 k,

$$\mathbf{N}_{TR}(t) = t * \mathbf{N}_{\text{const}} + \mathbf{N}_{\text{init}} \qquad \forall\ t \in \mathbb{N}. \tag{2}$$

The initial batch size of $\mathbf{N}_{\text{init}}$ was set to 1000 molecules and remained fixed for all increment schemes. The indices of molecules

in the initial batch were kept exactly the same while computing the learning curves to ensure comparable results.

Molecular structures were encoded with the many-body tensor representation (MBTR),[50] which has demonstrated superior accuracy in ML studies on molecular datasets.[51–53] The MBTR records atomic species, pairwise distances, and angles between atoms as components of a vector denoted by $k1$, $k2$, and $k3$, respectively. Previous work[15] has shown that omitting terms $k1$ and $k3$ results in a small loss in accuracy with molecular datasets but a large reduction in descriptor size. Consequently, we restricted our MBTR vectors to $k2$ terms only. For details on MBTR hyperparameter optimization, we refer the reader to our previous work.[15]

Since the MBTR vectors vary smoothly, they can be modeled well with GPs in kernel-based supervised learning regression. These non-parametric models access the entire training data in the form of a kernel matrix for the model to learn. While standard GPRs do not scale very well to large training datasets, scaling and data volume improvements can be achieved with sparse-GPs[54,55] or GPs trained and deployed on graphics processing units (GPUs).[56]

We trained the GP to predict molecular HOMO energy levels (target $\mathbf{y}$) based on the molecular representations (input $\mathbf{X}$). The trained model was then applied to previously unseen molecular structures $\mathbf{X}_*$, to compute the posterior predictive mean μ_{pred} (HOMO level prediction) and posterior predictive variance or uncertainty σ^2_{pred}. Given a data noise of σ_n^2,

$$\mu_{\text{pred}} = K(\mathbf{X}_*, \mathbf{X})\left[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}\right]^{-1}\mathbf{y}, \tag{3}$$

$$\sigma^2_{\text{pred}} = K(\mathbf{X}_*, \mathbf{X}_*) - K(\mathbf{X}_*, \mathbf{X})K_x^{-1}K(\mathbf{X}, \mathbf{X}_*). \tag{4}$$

In Eqs. (3) and (4), the symmetric kernel matrix K has individual matrix elements computed through the kernel function. Because the MBTR vector is smoothly varying, we selected the radial basis function (RBF) kernel (k_{RBF}),

$$K_x = K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}, \tag{5}$$

$$k_{RBF}(x, x') = \sigma_f \ \exp\left(\frac{-\|x - x'\|_2^2}{2l^2}\right). \tag{6}$$

In Eq. (6), the exponent can be expanded as follows:

$$\|x - x'\|_2 = \sqrt{\sum_i (x_i - x_i')^2}. \tag{7}$$

This is the Euclidean distance between the two MBTR vectors $\mathbf{x}$ and $\mathbf{x}'$. In Eq. (6), σ_f refers to a global scaling factor or signal variance in the GP literature,[46] and l represents the length scale.

We implemented the GPR in the Scikit-learn[57] (SKLearn) package and optimized the model hyperparameters by maximizing the *log marginal likelihood*,[46] which is a function of the training labels $\mathbf{y}$ and GP mean and variance computed on the training data [μ and σ^2 computed with $\mathbf{X}_* = \mathbf{X}$ in Eqs. (3) and (4)]. The length scale and output variance were initialized to 700 and 20 to match the hyperparameters from previously published[15] ML models trained on the same datasets. The hyperparameters were optimized over a range (bound of kernel hyperparameters in SKLearn) that varied four orders of magnitude, with the lower and upper hyperparameter search bounds set 100 times smaller or larger than the initial value. The GP prior mean was set to zero (`normalize_y = False` in SKLearn), and the number of optimizer restarts were set to 2. Given that the data labels were obtained with accurate DFT simulations, we set the model noise to a very low value of $\sigma_n^2 = 10^{-10}$ (unless otherwise stated). The optimal value of σ_n^2 was identified by performing a grid search on a log-scale from $[10^{-10}, 1]$. Using the default GPR and optimizing the noise on a grid (instead of using a White Kernel) enabled us to keep the GPR kernel as close as possible to the KRR kernel to ensure easy comparison of the hyperparameters with our previous work.[15] For each value of σ_n^2, we optimized the GPR log-marginal likelihood. For the optimized hyperparameter values, we computed the mean absolute error (MAE) on the test set. The grid search was repeated to find the hyperparameters with the lowest test set MAE.

To evaluate GPR model performance, we used both regression and classification metrics. To monitor regression, we computed the mean absolute error (MAE) of the model on the test set as a function of training set size. This allowed us to build ML learning curves and compare different AS approaches. In task 2, we gained further insight into the classification accuracy via the following classification metrics.[58] TPR is the ratio of the number of molecules correctly classified to be in-range (*true positive* or TP) and the total number of molecules in this class (*positives* or P). Similarly, FPR is the ratio of the number of molecules incorrectly classified to be in-range (false positive or FP) and the number of molecules that are not in the correct class (*negatives* or N). The TPR and FPR values range between 0 and 1. As the classification accuracy of the ML model improves, TPR tends toward 1 and FPR toward 0,

$$\begin{aligned} \text{TPR} &= \frac{TP}{P}, \\ \text{FPR} &= \frac{\text{FP}}{N}. \end{aligned} \tag{8}$$

IV. RESULTS

We began by identifying the best performing AS to compile maximally compact and informative datasets, as defined in task 1. The relative performance of different acquisition strategies (AS) was evaluated by comparing their learning curves. We explored the efficacy of different AS as a function of key method parameters: the acquisition batch size $\mathbf{N}_b$, which affects the growth rate of the training set, and the estimated data noise σ_n^2 in the GP model, which determines the smoothness of the model and, consequently, its performance. We also considered how the proposed ASs perform on different molecular datasets.

The first series of AS tests was carried out with the AA dataset because it was structurally the most redundant one. In our bid to actively learn the most compact training set, there was considerable similarity to be eliminated from the pool of AA molecules, and we expected to observe the largest difference in performance between the proposed AS schemes. Figure 4(a) presents the AS learning

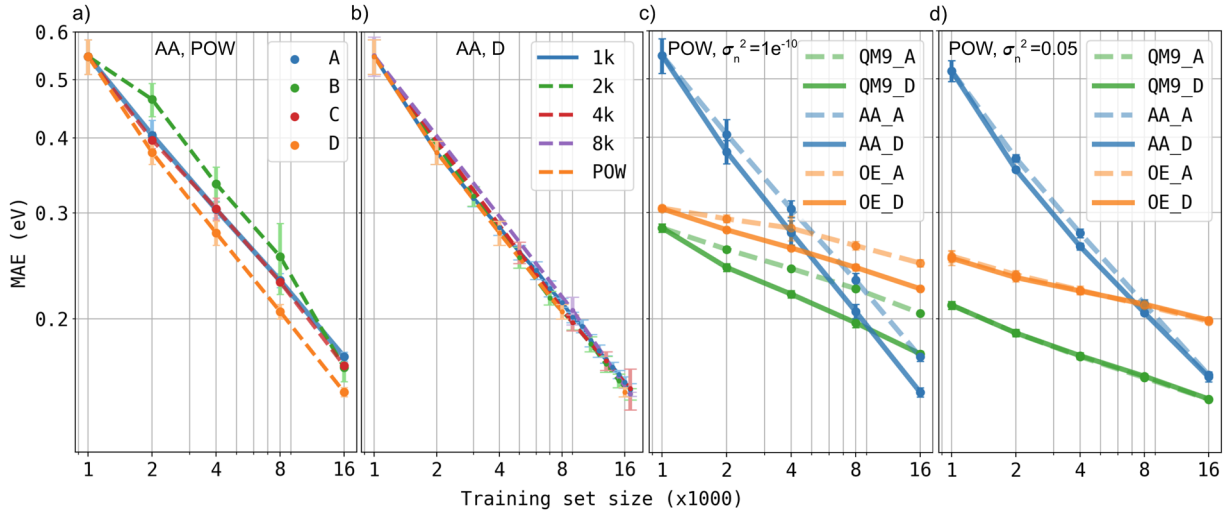

FIG. 4. AL learning curves in log–log scale for task 1, with test set MAEs computed from GP model predictions as a function of increasing training set size. (a) Performance of different AS for the AA dataset with the POW batch scheme. (b) Performance of different batch strategies for the AA dataset and AS D. (c) Strategy A and D performance on all datasets with $\sigma_n^2 = 10^{-10}$. (d) Strategy A and D performance on all datasets with $\sigma_n^2 = 0.05$. The data corresponding to the plots have been included in tables in Appendix B.

curves (averaged over 5 runs) computed with the POW $\mathbf{N}_b$ incrementing scheme for task 1, as the training set was compiled from 1 to 16 k molecules. Uncertainty-based strategy B was the worst AS, consistently achieving prediction errors higher than the random picking baseline A. At maximum training set size, we observed $\mathrm{MAE_B} = 0.164 \pm 0.001$ eV compared to $\mathrm{MAE_A} = 0.170 \pm 0.009$ eV. AS C, which clusters molecules based on their structural similarity, performed on par with the baseline, achieving an $\mathrm{MAE_C} = 0.166 \pm 0.001$ eV. Strategy D, which combines uncertainty and clustering, performed the best with $\mathrm{MAE_D} = 0.148 \pm 0.002$ eV. It was selected for the next set of tests. The error bars were obtained from five AL runs with the same GPR hyperparameters but different initial random seeds. Moreover, the reported error is caused by the variation over five runs in the GPR prediction mean, which is utilized to predict the HOMO values.

Batch sampling selects multiple molecules at a time, and while it allows AL to scale to larger datasets,[59] it is more susceptible to sampling redundant molecules. This is why we tested the sensitivity of strategy D to $\mathbf{N}_b$ choice. The POW incrementing scheme was compared to the constant $\mathbf{N}_b$ approach, with $\mathbf{N}_{\mathrm{const}}$ of 1, 2, 4, or 8 k. Results are presented in Fig. 4(b). Surprisingly, all sampling schemes exhibited the same ML performance. However, smaller constant $\mathbf{N}_b$ resulted in more AL iterations and was, consequently, slower to execute. We selected the POW scheme to proceed with because it covers a wide range of training set sizes with the fewest AL iterations.

Next, we considered whether strategy D performs equally well for different datasets. Figure 4(c) illustrates the learning curves for strategy D against random picking for datasets AA, QM9, and OE. The learning rates for the three datasets were very different, as indicated by our previous study.[15] In all cases, strategy D appeared to lower MAEs compared to the baseline, suggesting that similar

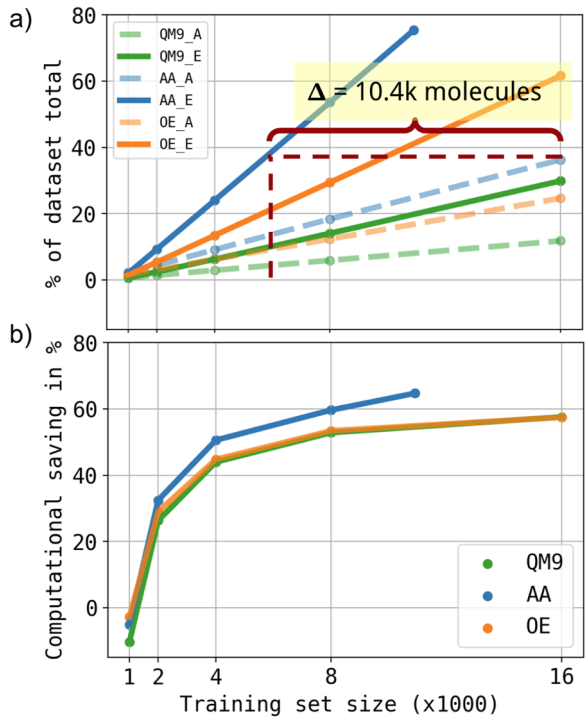

FIG. 5. AL model performance for task 2. (a) Number of correct structures (HOMO > ε) identified by AS A and E, presented as a percentage of the total in-range molecules in the dataset. (b) As shown in panel (a), AS E requires fewer training examples to achieve the same predictive accuracy as AS A. The plot presents the number of additional in-range molecules identified by AS E relative to AS A, expressed as a percentage of total in-range molecules in each dataset.

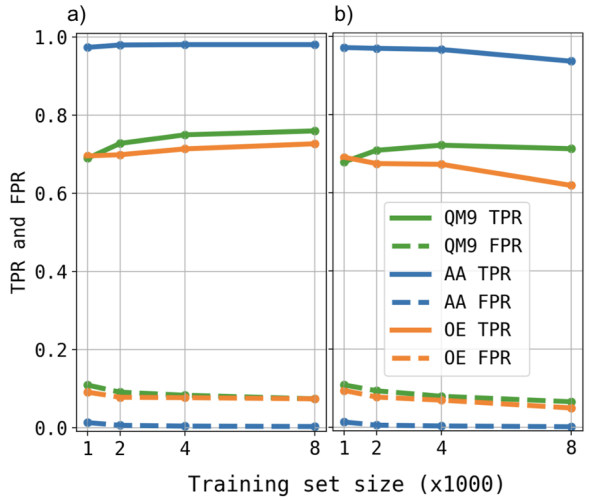

FIG. 6. Task 2 classification metrics TPR and FPR as a function of training set size, evaluated with strategy E for all 3 datasets on the (a) test set and (b) held-out set.

learning could be achieved with a smaller training set size. However, the MAEs for the baseline strategy A were consistently higher compared to the same data from previous work,[15] alerting us to a problem.

A careful review of our GPR revealed that our data noise settings $\sigma_n^2 = 10^{-10}$ were quite low and led to over-fitting. We conducted a grid search for optimal GP noise and found it to be as high as $\sigma_n^2 = 0.05$. To establish how data noise settings affect the performance of the AL, we recomputed the learning curves of strategies D and A for all datasets with the optimal σ_n^2 levels. The results in Fig. 4(d) now indicate that at higher noise levels, the performance of strategy D is indistinguishable from random picking A. Apart from minor savings for the redundant AA dataset, we discovered no benefit of active learning for task 1.

Next, we focus on task 2, where the objective is to identify molecules with HOMO energy greater than ε, merely based on structure. ε was chosen such that ~30% of the molecules in the datasets belong to this category. This resulted in $\varepsilon_{QM9} = -5.55$ eV, $\varepsilon_{AA} = -8.5$ eV, and $\varepsilon_{OE} = -5.2$ eV. Here, we inspect the performance of strategy E against strategy A for all datasets, with the POW batch scheme and optimal $\sigma_n^2 = 0.05$. Since this is a classification task, model performance can be evaluated by how many molecules of the correct class (HOMO in-range) are extracted by the AS from the total structural pool, verified against the computed HOMOs.

The learning curves in Fig. 5(a) illustrate the success of molecular classification as a function of the GP model training set size. Since QM9, AA and OE have different dataset sizes, the number of correct structures is presented as a percentage of the total in-range molecules in the dataset. Model performance increases linearly with the training set size, with a larger slope indicating a faster rate of extracting in-range molecules.

It is evident that AS E achieves systematically higher identification rates and extracts more correct structures compared to the random baseline. Different learning rates indicate that the molecules in the AA datasets were easiest to classify (over 70% correctly identified), followed by OE and QM9. Better AS E performance against the random baseline means that the same number of correct molecules could be identified in fewer iterations. For AA, 39% of the in-range molecules were identified with a training set size of 5600 with strategy E (dotted line), and the same percentage was achieved by the baseline with 16 000 training molecules. Less training set data translates to 10 400 fewer HOMO computations for the same model quality. We used the two learning curves to calculate the fraction of relative computational savings produced by strategic sampling. The data are illustrated in Fig. 5(b). The best savings could be obtained for the AA dataset (64%), while the savings for OE and QM9 were surprisingly similar (57%).

Next, we inspect the classification accuracy of AS E. Figure 6 illustrates the rate of TP and FP classifications as a function of training set size. The metrics computed on a randomly chosen test set in Fig. 6(a) assess the objective improvement in classification accuracy. It is interesting to observe that the model classifies well already with

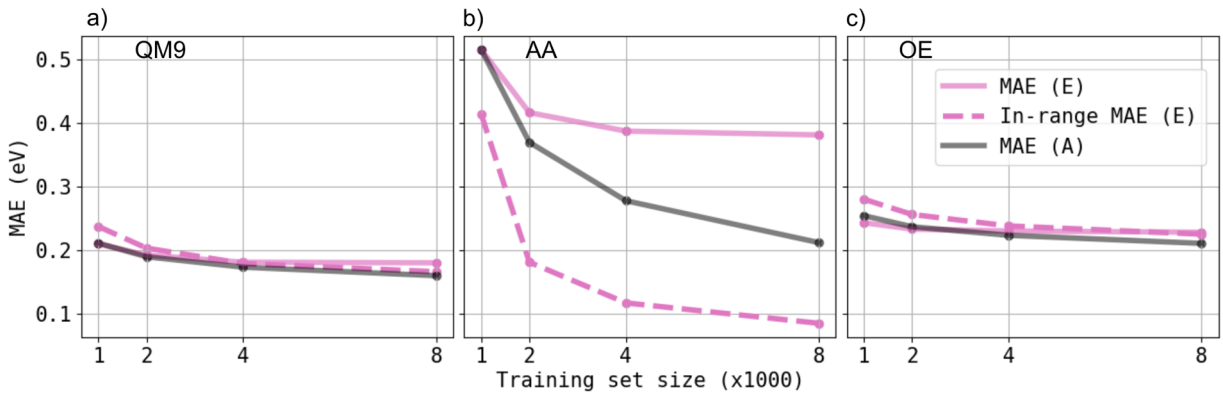

FIG. 7. Task 2 regression MAE metrics evaluated on (a) QM9, (b) AA, and (c) OE datasets. MAE evaluated with AS E on the test set (pink solid line) and a test set of in-range molecules (pink dashed line) are compared with those evaluated with AS A on the test set (black solid line).

little training data. The improvement in TP with more data is very slight, almost none for the AA dataset. The FP wrong classifications register a very small decrease on the test set.

We also explore the metrics on the held-out set in Fig. 6(b) to establish if the classification accuracy changes as the held-out set is depleted of relevant molecules with subsequent iterations. There is a slight overall decrease in accuracy, which suggests that as more relevant molecules are moved from the held-out set to the training set, the model cannot classify the remaining structures as well. This trend is most notable in TP data for the AA and OE datasets, where structural diversity is high. The drop in accuracy leads to fewer relevant molecules being added to the training set in the later stages of AL.

In this task, the classification accuracy of the model acutely depends on the quality of supervised regression: the prediction of the HOMO energy based on molecular structure. Figure 7 presents the evolution of prediction MAE with training set size for all datasets. Overall, we observe the prediction errors decreasing, as expected. The results are most notable for the AA dataset in Fig. 7(b). Here, the prediction MAEs on the in-range molecules are reduced dramatically compared to the errors on the random subset. This is not the case for the QM9 and OE datasets, where the MAE of predictions on the in-range and randomly selected molecules are nearly identical.

V. DISCUSSION

Among the two proposed active learning objectives, task 1 is the most commonly studied in the literature.[19,20,60,61] We carried out a comparative study of AL strategies to explore which selection criteria compile superior training sets for the best predictive models [see Fig. 4(a)]. We observed that combining prediction uncertainty with clustering molecular structures (strategy D) yields the best results (lowest test set MAE), which corroborates similar findings in the literature.[19,20]

In this study, uncertainty alone was a sub-optimal criterion and consistently yielded higher MAEs compared to all other strategies. This, however, contradicts other findings in the literature.[60,61] Such a discrepancy can be explained by two key differences: the choice of machine learning model and the optimization task. In this GP-based study, uncertainty is modeled differently[62] than with the neural networks employed in earlier work.[61] Similarly, the focus here is on predicting HOMO energy value instead of inter-atomic potentials.[60] Minimizing the HOMO prediction error belongs to task type 1, while training an inter-atomic potential is of task type 2, as alluded to later in this section. For this reason, the same uncertainty-based acquisition strategy performs very differently in the two cases. Both these factors could lead to different learning outcomes, a factor to which we will return later in this section.

Despite previous work with different batch sizes in AL,[63] there is no insight into how batch size affects the quality of the GP models. Here, we compare constant and adaptive batch schemes and observe that batch size has a negligible effect on the performance of the best acquisition strategy.

In contrast, dataset noise (σ_n^2) has a significant impact on the performance of AL. If the GP overfits on the training data for low σ_n^2 values, the test set MAE increases, as observed for the random strategy (see AS A in panels c and d of Fig. 4). AL can then provide a benefit by balancing the dataset through clustered uncertainty minimization [compare AS A and D in Fig. 4(c)]. Strategy D thus ensures a higher diversity than random sampling, which reduces overfitting. The GPR of strategy D subsequently generalizes better to the test set, and the MAE drops faster. For larger noise values, the accuracy of the GP increases as overfitting on the training set reduces, as observed in Fig. 4(d). The AL benefit disappears since the best strategy to resemble the randomly drawn test set also in the training set is to randomly assemble it. In future research, strategies C and D could be further improved by incorporating an updated loss function weighted by the cluster sizes.

There are conflicting reports in the literature on the benefits of AL in dataset compilation. However, a closer look reveals that benefits depend on the prediction task and, more specifically, on the bounds of the search space sampled by AL. Training interatomic potentials is a prototypical example of an unbounded search task, where atomic configurations are sampled from a near-infinite pool of possible structures. Since atoms in these tasks can explore real-space continuously, there is no limit to the number of possible structures to pick. The test set, however, is constrained to a certain part of real space governed by the laws of quantum mechanics, namely, the vicinity of equilibrium structures (interatomic potentials) or the surroundings of a molecular dynamics trajectory. This generates an intrinsic difference between the distributions of structures in the training and test sets. AL can then efficiently reduce the number of training structures as it aligns the training with the test set.[29]

This work deals with dataset compilation from a large but finite pool of molecular structures. This is standard for property prediction tasks, in the absence of generative models to open up the search space bounds. Here, mere random picking already generates a matching distribution of the training and test sets, as illustrated in Fig. 8(a). AL sampling can hardly offer any benefit in re-sampling the training set, since there is no difference to bridge. Our finding in task 1 is, therefore, consistent with previous work.[63] In task 2, however, targeting molecules with particular HOMO values translates to a target distribution shifted away from the general structure pool [Fig. 8(b)]. The HOMO > ε target constitutes the upper third of the

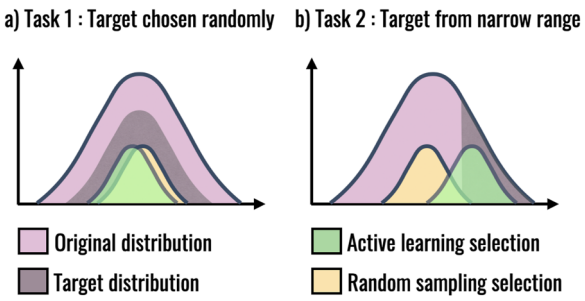

FIG. 8. Illustration of the two active learning tasks. (a) In task 1, the target is selected randomly from the entire original distribution; consequently, random sampling achieves a good representation of the target, and active learning provides no benefit over random sampling. (b) In task 2, the target is selected from a narrow region of the original distribution; here, active learning can adapt and represent the target well, outperforming random sampling.

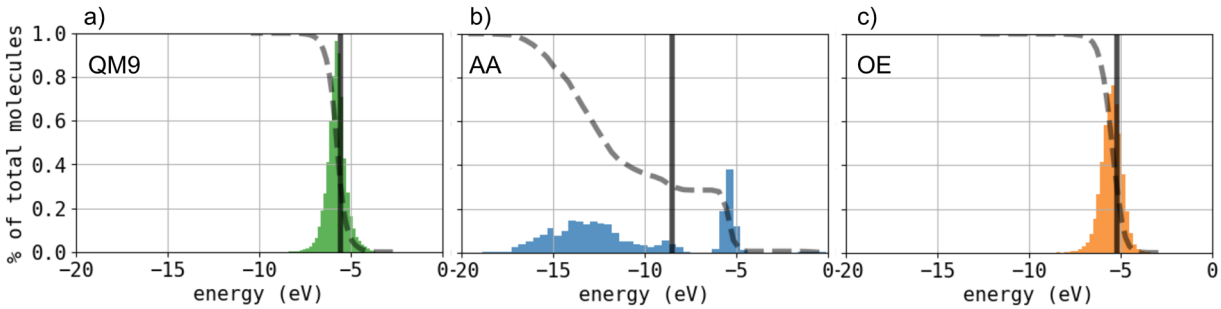

FIG. 9. Distribution of HOMO energy values for (a) QM9, (b) AA, and (c) OE. The vertical line indicates the classification boundary (ε), and the dashed line indicates the cumulative sum of the distribution of molecules. The classification boundary roughly includes 30% of the molecules from the dataset. This includes molecules with HOMO greater than −5.55, −8.5, and −5.2 eV for QM9, AA, and OE, respectively.

label distribution, which would be sampled poorly by the random strategy A. For such a use case, AL quickly focuses on the appropriate structures and provides a distinct benefit. It follows that the benefits of AL can be observed only in the case of a shift between the structural distributions of the test set and the overall search space. When the structure pool is finite, test set engineering can be used to fabricate a shift.

Targeted property search presents an example of intrinsic test set engineering, where AL can yield computational benefits. Strategy E was able to identify more desirable molecules in fewer AL iterations in comparison to random selection. This occurs because strategy E biased the training set with more molecules in the correct HOMO category, which improves the predictive accuracy of the model in the target HOMO range. Such a strategy of biasing the training set can be detrimental to the learning process.[30,31,64] However, when used judiciously for a specific task, biasing the training data has been shown to improve model performance.[13,28,65,66]

We observed that AL benefits can vary considerably with dataset type. The magnitude of computational savings identified in Fig. 5 arises directly from the classification accuracy observed in Fig. 6. For the AA dataset, near-perfect classification was observed even for small training sets. Accuracy for the QM9 and OE datasets improved slowly but converged to a constant value below 0.8 in TPR. To interpret this, we review the distribution of HOMO labels in Fig. 9 of the Appendix with regard to the position of the ε property boundary for correct classification. QM9 and OE boast an unimodal HOMO distribution. Since the decision boundary is placed near the distribution mode, many molecular structures are vulnerable to misclassification already for small HOMO prediction errors. This ultimately limits the classification accuracy, even with large training sets. In the bimodal distribution of HOMOs for the AA dataset, the decision boundary includes much of the relevant peak, which is why the initial accuracy is high. The subsequent addition of structures from the second mode of low HOMO values cannot improve this classification training set. AA classification accuracy is, therefore, consistently high, and that translates to good selectivity and large-scale computational savings.

The position of the decision boundary within the HOMO distribution also explains the quality of supervised regression in Fig. 7. For QM9 and OE datasets, the boundary near the HOMO label mode means that the molecules on either side of the classification boundary are similar. The randomly drawn dataset and the AL dataset capture similar information. For the bi-modal distribution of AA HOMO values, the AL model is primarily built on molecules from one mode of the distribution, whereas the random model contains both. Since the lower HOMO peak contributes little useful structural information to the model, AL performance is notably better.

As the GPR models grow more accurate, it is interesting that the classification accuracy on the held-out set molecules in Fig. 6(b) is reduced for all datasets. This indicates that successive AL iterations deplete the held-out set of molecules that are relevant and easy to classify. The GPR model accuracy is improved, but because the remaining structures are harder to classify, the net effect is the decrease of TPR for all three datasets.

VI. CONCLUSION

In this study, we proposed novel applications and presented a systematic analysis of AL methodology in molecular and materials science. The objectives were to compile compact and maximally informative datasets or identify molecules with targeted properties with the fewest calculations performed. The performance of the proposed algorithms was analyzed to identify the best settings to employ AL. Our results revealed that, for finite size datasets deployed in this work, AL provides no benefits for minimizing global error metrics such as the MAE. Instead, we found that computational savings achieved with AL are dependent on the distribution of target molecules in the task with respect to the total dataset distribution. These observations help to reconcile seemingly contradictory reports in the AL literature.

For applications minimizing global errors such as MAE, the target distribution is drawn randomly and resembles the parent distribution. Here, the best AL strategy is to draw randomly, instead of using criteria based on uncertainty and diversity. Our findings indicate that AL provides significant computational savings in applications where the target molecules are drawn from a narrow region of the dataset. For targeted property search, the proposed AL strategy provided significant computational savings of up to 64% as compared to random sampling. The savings can vary considerably with the compound space sampled. Searching for

molecular structures with targeted properties, therefore, presents a useful application in materials research, where AL delivers the most benefit.

ACKNOWLEDGMENTS

This study received the financial support from the Academy of Finland through its flagship program, the Finnish Center for Artificial Intelligence, and the Centers of Excellence Program (CoE VILMA, Grant No. 346377). Computing resources from the Aalto Science-IT project and the CSC—IT Center for Science, Finland, are gratefully acknowledged. In addition, K.G. thanks the Finnish Cultural Foundation (Grant No. 00210309) for funding the research.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Ethics Approval

Ethics approval is not required.

Author Contributions

M.T. and P.R. conceived the initial idea. K.G. implemented the machine learning model and ran the experiments. All the authors contributed to writing the manuscript.

TABLE I. Data to reproduce Fig. 4(a). The table contains MAE values (in eV) averaged over 5 runs and the corresponding standard deviation.

	Figure 4(a) (POW, $\sigma_n^2 = 10^{-10}$)			
	AA			
Training set size (×1000)	A	B	C	D
1	0.547 ± 0.036	0.547 ± 0.036	0.547 ± 0.036	0.547 ± 0.036
2	0.405 ± 0.024	0.464 ± 0.030	0.397 ± 0.012	0.379 ± 0.017
4	0.304 ± 0.009	0.336 ± 0.022	0.305 ± 0.013	0.278 ± 0.013
8	0.232 ± 0.006	0.254 ± 0.034	0.231 ± 0.002	0.206 ± 0.006
16	0.173 ± 0.003	0.166 ± 0.008	0.167 ± 0.002	0.151 ± 0.003

TABLE II. Data to reproduce Fig. 4(b). The table contains MAE values (in eV) averaged over 5 runs and the corresponding standard deviation.

	Figure 4(b) (D, $\sigma_n^2 = 10^{-10}$)				
	AA				
Training set size (×1000)	1 k	2 k	4 k	8 k	POW
1	0.547 ± 0.036	0.547 ± 0.036	0.547 ± 0.036	0.548 ± 0.041	0.547 ± 0.036
2	0.378 ± 0.017				0.151 ± 0.003
3	0.318 ± 0.010	0.320 ± 0.012			
4	0.284 ± 0.008				0.151 ± 0.003
5	0.258 ± 0.007	0.253 ± 0.010	0.258 ± 0.011		
6	0.241 ± 0.008				
7	0.225 ± 0.007	0.217 ± 0.006			
8	0.213 ± 0.004				0.151 ± 0.003
9	0.201 ± 0.004	0.197 ± 0.005	0.197 ± 0.006	0.207 ± 0.011	
10	0.195 ± 0.005				
11	0.184 ± 0.004	0.182 ± 0.005			
12	0.177 ± 0.004				
13	0.171 ± 0.002	0.168 ± 0.005	0.170 ± 0.006		
14	0.166 ± 0.003				
15	0.162 ± 0.003	0.158 ± 0.004			
16	0.157 ± 0.003				0.151 ± 0.003
17	0.153 ± 0.004	0.150 ± 0.003	0.153 ± 0.012	0.150 ± 0.001	

TABLE III. Data to reproduce Fig. 4(c). The table contains MAE values (in eV) averaged over 5 runs and the corresponding standard deviation.

	Figure 4(c) (POW, $\sigma_n^2 = 10^{-10}$)					
	QM9		AA		OE	
Training set size (×1000)	A	D	A	D	A	D
1	0.283 ± 0.004	0.283 ± 0.004	0.547 ± 0.036	0.547 ± 0.036	0.305 ± 0.003	0.305 ± 0.003
2	0.261 ± 0.003	0.243 ± 0.003	0.405 ± 0.024	0.379 ± 0.017	0.293 ± 0.004	0.281 ± 0.002
4	0.242 ± 0.002	0.220 ± 0.003	0.304 ± 0.010	0.278 ± 0.013	0.283 ± 0.013	0.262 ± 0.003
8	0.224 ± 0.002	0.197 ± 0.003	0.231 ± 0.006	0.206 ± 0.006	0.264 ± 0.004	0.243 ± 0.003
16	0.204 ± 0.001	0.175 ± 0.001	0.173 ± 0.003	0.151 ± 0.003	0.247 ± 0.003	0.224 ± 0.002

TABLE IV. Data to reproduce Fig. 4(d). The table contains MAE values (in eV) averaged over 5 runs and the corresponding standard deviation.

	Figure 4(d) (POW, $\sigma_n^2 = 0.5$)					
	QM9		AA		OE	
Training set size (×1000)	A	D	A	D	A	D
1	0.211 ± 0.003	0.211 ± 0.003	0.515 ± 0.021	0.515 ± 0.021	0.252 ± 0.007	0.252 ± 0.007
2	0.189 ± 0.002	0.189 ± 0.003	0.369 ± 0.005	0.353 ± 0.003	0.234 ± 0.004	0.234 ± 0.004
4	0.173 ± 0.002	0.174 ± 0.002	0.278 ± 0.004	0.263 ± 0.004	0.222 ± 0.003	0.222 ± 0.003
8	0.159 ± 0.001	0.160 ± 0.001	0.212 ± 0.003	0.204 ± 0.002	0.211 ± 0.002	0.211 ± 0.002
16	0.147 ± 0.001	0.147 ± 0.001	0.161 ± 0.002	0.160 ± 0.002	0.199 ± 0.001	0.199 ± 0.001

Kunal Ghosh: Conceptualization (equal); Investigation (equal); Methodology (equal); Software (equal); Writing – original draft (equal). **Milica Todorović**: Conceptualization (equal); Data curation (equal); Investigation (equal); Methodology (equal); Resources (equal); Supervision (equal); Writing – original draft (equal). **Aki Vehtari**: Methodology (supporting); Supervision (supporting); Writing – original draft (equal). **Patrick Rinke**: Conceptualization (equal); Funding acquisition (lead); Supervision (equal); Writing – original draft (equal).

DATA AVAILABILITY

The data that support the findings of this study are openly available in Zenodo, record numbers 3967308,[67] 4035923,[68] and 4035918.[69] The code to reproduce the results in this publication can be found on GitHub.[70]

APPENDIX A: DISTRIBUTION OF HOMO ENERGY VALUES

Figure 9 shows the distribution of HOMO energy values for (a) QM9, (b) AA, and (c) OE.

APPENDIX B: DATA TO REPRODUCE FIG. 4

Data to reproduce Figs. 4(a)–4(d) can be found in Tables I–IV.

REFERENCES

[1] H. Kulik, T. Hammerschmidt, J. Schmidt, S. Botti, M. A. L. Marques, M. Boley, M. Scheffler, M. Todorović, P. Rinke, C. Oses, A. Smolyanyuk, S. Curtarolo, A. Tkatchenko, A. Bartok, S. Manzhos, M. Ihara, T. Carrington, J. Behler, O. Isayev, M. Veit, A. Grisafi, J. Nigam, M. Ceriotti, K. T. Schütt, J. Westermayr, M. Gastegger, R. Maurer, B. Kalita, K. Burke, R. Nagai, R. Akashi, O. Sugino, J. Hermann, F. Noé, S. Pilati, C. Draxl, M. Kuban, S. Rigamonti, M. Scheidgen, M. Esters, D. Hicks, C. Toher, P. Balachandran, I. Tamblyn, S. Whitelam, C. Bellinger, and L. M. Ghiringhelli, Electron. Struct. **4**, 023004 (2022).

[2] L. Himanen, A. Geurts, A. S. Foster, and P. Rinke, Adv. Sci. **6**, 1900808 (2019).

[3] A. P. Bartók, S. De, C. Poelking, N. Bernstein, J. R. Kermode, G. Csányi, and M. Ceriotti, Sci. Adv. **3**, e1701816 (2017).

[4] A. Merchant, S. Batzner, S. S. Schoenholz, M. Aykol, G. Cheon, and E. D. Cubuk, Nature **624**, 80–85 (2023).

[5] A. G. Kusne, H. Yu, C. Wu, H. Zhang, J. Hattrick-Simpers, B. DeCost, S. Sarker, C. Oses, C. Toher, S. Curtarolo, A. V. Davydov, R. Agarwal, L. A. Bendersky, M. Li, A. Mehta, and I. Takeuchi, Nat. Commun. **11**, 5966 (2020).

[6] F. Ren, L. Ward, T. Williams, K. J. Laws, C. Wolverton, J. Hattrick-Simpers, and A. Mehta, Sci. Adv. **4**, eaaq1566 (2018).

[7] P. Domingos, Commun. ACM **55**, 78–87 (2012).

[8] K. T. Schütt, P.-J. Kindermans, H. E. Sauceda, S. Chmiela, A. Tkatchenko, and K.-R. Müller, in *Neural Information Processing Systems* (Curran Associates, Inc., 2017), Vol. 31, pp. 992–1002.

[9] K. Ghosh, A. Stuke, M. Todorović, P. B. Jørgensen, M. N. Schmidt, A. Vehtari, and P. Rinke, Adv. Sci. **6**, 1801367 (2019).

[10] J. Westermayr and R. J. Maurer, Chem. Sci. **12**, 10755 (2021).

[11] K. Atz, F. Grisoni, and G. Schneider, Nat. Mach. Intell. **3**, 1023 (2021).

[12] J. Behler, Int. J. Quantum Chem. **115**, 1032 (2015).

[13] J. Westermayr, J. Gilkes, R. Barrett, and R. J. Maurer, Nat. Comput. Sci. **3**, 139 (2023).

[14]R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld, Sci. Data **1**, 140022 (2014).
[15]A. Stuke, M. Todorović, M. Rupp, C. Kunkel, K. Ghosh, L. Himanen, and P. Rinke, J. Chem. Phys. **150**, 204121 (2019).
[16]B. Settles, *Active Learning* (Springer International Publishing, 2012).
[17]Y. Fujiwara, Y. Yamashita, T. Osoda, M. Asogawa, C. Fukushima, M. Asao, H. Shimadzu, K. Nakao, and R. Shimizu, J. Chem. Inf. Model. **48**, 930–940 (2008).
[18]D. Reker, P. Schneider, and G. Schneider, Chem. Sci. **7**, 3919 (2016).
[19]K. Zhou, K. Wang, J. Tang, J. Feng, B. Hooi, P. Zhao, T. Xu, and X. Wang, LoG **198**, 29 (2022).
[20]S. Hwang, J. Choi, and J. Choi, IEEE Access **10**, 110983 (2022).
[21]B. Desai, K. Dixon, E. Farrant, Q. Feng, K. R. Gibson, W. P. van Hoorn, J. Mills, T. Morgan, D. M. Parry, M. K. Ramjee, C. N. Selway, G. J. Tarver, G. Whitlock, and A. G. Wright, J. Med. Chem. **56**, 3033 (2013).
[22]J. Besnard, G. F. Ruda, V. Setola, K. Abecassis, R. M. Rodriguiz, X.-P. Huang, S. Norval, M. F. Sassano, A. I. Shin, L. A. Webster, F. R. Simeons, L. Stojanovski, A. Prat, N. G. Seidah, D. B. Constam, G. R. Bickerton, K. D. Read, W. C. Wetsel, I. H. Gilbert, B. L. Roth, and A. L. Hopkins, Nature **492**, 215 (2012).
[23]A. W. Naik, J. D. Kangas, D. P. Sullivan, and R. F. Murphy, eLife **5**, 10047 (2016).
[24]S. Viet Johansson, H. Gummesson Svensson, E. Bjerrum, A. Schliep, M. Haghir Chehreghani, C. Tyrchan, and O. Engkvist, Mol. Inf. **41**, 2200043 (2022).
[25]S. Vandenhaute, M. Cools-Ceuppens, S. DeKeyser, T. Verstraelen, and V. Van Speybroeck, npj Comput. Mater. **9**, 19 (2023).
[26]Y. Wen, Z. Li, Y. Xiang, and D. Reker, Digital Discovery **4**, 1134 (2023).
[27]A. Jose, J. P. A. de Mendonça, E. Devijver, N. Jakse, V. Monbet, and R. Poloni, Data Min. Knowl. Discovery **38**, 420–460 (2023).
[28]V. Besel, M. Todorović, T. Kurtén, H. Vehkamäki, and P. Rinke, J. Aerosol Sci. **179**, 106375 (2024).
[29]V. Zaverkin and J. Kästner, Mach. Learn.: Sci. Technol. **2**, 035009 (2021).
[30]J. W. Richards, D. L. Starr, H. Brink, A. A. Miller, J. S. Bloom, N. R. Butler, J. Berian James, J. P. Long, and J. Rice, Astrophys. J. **744**, 192 (2011).
[31]S. Farquhar, Y. Gal, and T. Rainforth, in International Conference on Learning Representations, 2021.
[32]J. Gonzalez, Z. Dai, P. Hennig, and N. Lawrence, in *Proceedings of Machine Learning Research* (PMLR, 2016), Vol. 51, p. 648.
[33]A. Kirsch, J. van Amersfoort, and Y. Gal, in *Advances in Neural Information Processing Systems* (Curran Associates, Inc., 2019), Vol. 32, pp. 7026–7037.
[34]M. Ropo, M. Schneider, C. Baldauf, and V. Blum, Sci. Data **3**, 160009 (2016).
[35]J. P. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett. **78**, 1396 (1997).
[36]A. Tkatchenko and M. Scheffler, Phys. Rev. Lett. **102**, 073005 (2009).
[37]S. De, A. P. Bartók, G. Csányi, and M. Ceriotti, Phys. Chem. Chem. Phys. **18**, 13754 (2016).
[38]N. Artrith, A. Urban, and G. Ceder, Phys. Rev. B **96**, 014112 (2017).
[39]S. De, F. Musil, T. Ingram, C. Baldauf, and M. Ceriotti, J. Cheminf. **9**, 6 (2017).
[40]L. Ruddigkeit, R. van Deursen, L. C. Blum, and J.-L. Reymond, J. Chem. Inf. Model. **52**, 2864 (2012).
[41]A. Stuke, C. Kunkel, D. Golze, M. Todorović, J. T. Margraf, K. Reuter, P. Rinke, and H. Oberhofer, Sci. Data **7**, 58 (2020).
[42]F. H. Allen, Acta Crystallogr., Sect. B **58**, 380 (2002).
[43]C. Schober, K. Reuter, and H. Oberhofer, J. Phys. Chem. Lett. **7**, 3973 (2016).
[44]C. Schober, Ph.D. dissertation (TU München, 2017).
[45]J. Y. Choi, P. Zhang, K. Mehta, A. Blanchard, and M. Lupo Pasini, J. Cheminf. **14**, 70 (2022).
[46]C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning* (MIT Press, 2006).
[47]A. Wang, H. Liang, A. McDannald, I. Takeuchi, and A. G. Kusne, Oxford Open Mater. Sci. **2**, itac006 (2022).
[48]L. Bassman Oftelie, P. Rajak, R. K. Kalia, A. Nakano, F. Sha, J. Sun, D. J. Singh, M. Aykol, P. Huck, K. Persson, and P. Vashishta, npj Comput. Mater. **4**, 74 (2018).
[49]D. Reker, *Artificial Intelligence in Drug Discovery* (The Royal Society of Chemistry, 2020).
[50]H. Huo and M. Rupp, Mach. Learn.: Sci. Technol. **3**, 045017 (2022).
[51]O. Rahaman and A. Gagliardi, J. Chem. Inf. Model. **60**, 5971 (2020).
[52]M. P. Bahlke, N. Mogos, J. Proppe, and C. Herrmann, J. Phys. Chem. A **124**, 8708 (2020).
[53]E. Lumiaro, M. Todorović, T. Kurten, H. Vehkamäki, and P. Rinke, Atmos. Chem. Phys. **21**, 13227 (2021).
[54]E. Snelson and Z. Ghahramani, in *Advances in Neural Information Processing Systems* (MIT Press, 2005), Vol. 18.
[55]J. Hensman, A. G. Matthews, M. Filippone, and Z. Ghahramani, "MCMC for variationally sparse Gaussian processes," in *Advances in Neural Information Processing Systems* (Curran Associates, Inc., 2015), Vol. 28, https://papers.nips.cc/paper_files/paper/2015/hash/6b180037abbebea991d8b1232f8a8ca9-Abstract.html.
[56]K. Wang, G. Pleiss, J. Gardner, S. Tyree, K. Q. Weinberger, and A. G. Wilson, in *Advances in Neural Information Processing Systems* (Curran Associates, Inc., 2019), Vol. 32.
[57]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, J. Mach. Learn. Res. **12**, 2825 (2011).
[58]T. Fawcett, Pattern Recognit. Lett. **27**, 861 (2006).
[59]G. Citovsky, G. DeSalvo, C. Gentile, L. Karydas, A. Rajagopalan, A. Rostamizadeh, and S. Kumar, in *NeurIPS* (JMLR.org, 2021), Vol. 34, p. 11933.
[60]M. Kulichenko, K. Barros, N. Lubbers, Y. W. Li, R. Messerly, S. Tretiak, J. S. Smith, and B. Nebgen, Nat. Comput. Sci. **3**, 230 (2023).
[61]Y. Zhang and A. A. Lee, Chem. Sci. **10**, 8154 (2019).
[62]Y. Li, S. Rao, A. Hassaine, R. Ramakrishnan, D. Canoy, G. Salimi-Khorshidi, M. Mamouei, T. Lukasiewicz, and K. Rahimi, Sci. Rep. **11**, 20685 (2021).
[63]V. Zaverkin, D. Holzmüller, I. Steinwart, and J. Kästner, Digital Discovery **1**, 605 (2022).
[64]Z. Hengrui, C. Wei, J. M. Rondinelli, and W. Chen, Appl. Phys. Rev. **10**, 021403 (2023).
[65]L. Filstroff, I. Sundin, P. Mikkola, A. Tiulpin, J. Kylmäoja, and S. Kaski, arXiv:2106.04193 (2021).
[66]I. Sundin, P. Schulam, E. Siivola, A. Vehtari, S. Saria, and S. Kaski, in *International Conference on Machine Learning* (PMLR, 2019), p. 6046.
[67]K. Ghosh (2020). "MBTR AA," Zenodo. https://doi.org/10.5281/zenodo.3967308.
[68]K. Ghosh (2020). "MBTR OE62," Zenodo. https://doi.org/10.5281/zenodo.4035923.
[69]K. Ghosh (2020). "MBTR QM9," Zenodo. https://doi.org/10.5281/zenodo.4035918.
[70]See https://github.com/kunalghosh/Multi_Fidelity_Prediction_GP/tree/testing_runs for the code utilized to replicate the results presented in this publication.