A diversity maximizing active learning strategy for graph neural network models of chemical properties

Journal:	*Molecular Systems Design & Engineering*
Manuscript ID	ME-ART-04-2022-000073.R2
Article Type:	Paper
Date Submitted by the Author:	17-Aug-2022
Complete List of Authors:	Li, Bowen; Lehigh University, chemical and biomolecular enigneering Rangarajan, Srinivas; Lehigh University,

All articles must include a separate 'Design, System, Application' statement. This statement should not be a summary of the work reported as in the article abstract but should be a paragraph of no more than 200 words that:

Explains the molecular design or optimisation strategy and its general utility.
Emphasizes the desired systems functionality and design constraints.
Highlights the immediate or future application potential of the work.

Design statement

We proposed an active learning algorithm to reduce the budget to train an accurate deep learning model for molecular property prediction. It identifies a subset of a molecule library to train a model that accurately predicts the remaining molecules. The process is performed by iteratively updating the training set with a batch of molecules that maximize diversity in the latent space. In general, it can be applied to material design or reaction network annotation.

The desired system's functionality and constraints mainly depend on the model representation. In our benchmarking case with an atomistic deep learning model. It turns out that our algorithm performs well in a library of diverse atom types and sizes but not in a library that constitutes of few atoms type while having a wide range of functional groups.

This method is particularly valuable in situations where acquiring labeled data is expensive and carefully training dataset can enable building accurate models with a data budget.

Journal Name

ARTICLE TYPE

Cite this: DOI: 00.0000/xxxxxxxxxx

A diversity maximizing active learning strategy for graph neural network models of chemical properties†

Bowen Li [a] and Srinivas Rangarajan*[a]

Received Date
Accepted Date

DOI: 00.0000/xxxxxxxxxx

This paper presents a diversity-maximizing strategy for actively constructing a compact molecule set for training graph neural network molecular property models. In particular, we consider the core-set selection problem, viz., finding a training set S that is (1) representative and (2) a subset of a pre-defined space U of interest. The strategy iteratively adds new molecules into S so that the its diversity is maximized (in a greedy way) with respect to U; the diversity itself is determined from a Euclidean distance metric of a feature vector that is extracted from the graph neural network model at that iteration. We apply this strategy to retrospectively construct compact training sets for a number of experimental and computed molecular properties and show that it outperforms random sampling of U in almost all cases. Random sampling and the proposed active learning strategy, however, perform similarly for the QM7 (computed heat of atomization) dataset; further inspection using data visualization and analysis indicates that this is attributable to the manner in which the molecule set was created to maximize functional group diversity. Our method, in general, is property agnostic and does not require the calculation of prediction uncertainty at each iteration.

1 Introduction

Accurate property estimation is required for the modeling and design of many molecular systems. Examples include the design of chemicals,[1–5] energy carriers,[6,7] and drugs,[8–10] and multiscale modeling of reaction systems such as combustion of hydrocarbons[11] or catalytic valorization of biomass,[12,13] wherein fast and reliable values of molecular properties (*e.g.* heats or entropy of formation, boiling and melting points, *etc.*) are required to compute the properties of (or screen) a vast number of molecules and intermediates. While conventional methods such as experiments or quantum chemical calculations allow for calculating these properties reliably, they are not fast enough for computational tractability.

Machine learning, and in particular deep learning, has quickly become the approach of choice to compute properties for a large space of molecules efficiently, particularly in the context of molecule design and discovery.[14–21] However, the quality of machine learning models is dependent on the underlying fidelity and quantity of the training data. Deep learning models, that offer near chemical accuracy for many properties, rely on a training dataset of tens (if not hundreds) of thousands of data points. Acquiring such large datasets, computational or experimental, at high fidelity can be prohibitively expensive in many cases; this consequently limits our ability to apply deep learning strategies to such problems. Since, arguably, not all data points are equally informative[22],a careful selection of training points can, in principle, reduce the cost of acquiring training data while still building reliable models.[23–26]

Active learning is an iterative strategy of selecting training data to reduce the cost of gathering data while achieving a good performance with a compact model, especially for the fields where each data point is resource-consuming to acquire, including in computer vision[27–29], material discovery[30–32], molecular simulations[33–36], and reaction design[37]. Bayesian optimization, often employed in molecular discovery, also requires active learning of a model while simultaneously maximizing/minimizing a desired property.[38,39] These active learning approaches usually rely on quantifying the uncertainty of the points in the molecular or material space and are designed to eliminate regions with large prediction uncertainties. Methods to quantify the uncertainty include (1) calculating the variance between multiple predictions obtained by training an ensemble of neural network models[40] or many instances from one model wherein at each training step each neuron weight is randomly set to zero with a probability, (2) using Bayesian approaches (*e.g.* Bayesian neural networks),[41,42] wherein the parameters are assumed to be Gaussian variables and the final prediction for each point is represented by a distribution with a mean and variance (which represents uncertainty of the model).[43] (3) distance-based similarity methods wherein the distance between known and new molecules captured via latent

[a]*Department of Chemical and Biomolecular Engineering, Lehigh University, 111 Research Dr, Bethlehem, 18015, PA, USA; E-mail: srr516@lehigh.edu*
[b] *Address, Address, Town, Country.*
† Electronic Supplementary Information (ESI) available: [details of any supplementary information available should be included here]. See DOI: 00.0000/00000000.

features is related to uncertainty[44,45] and (4) pipeline methods wherein two models are trained subsequently to provide uncertainties;[46,47] the first model is typically a supervised neural network model, and the second model (a Gaussian process model) uses the latent features obtained from the first model to provide predictions and uncertainties.

We here consider a version of the problem of core set selection for training graph neural network models of molecular properties: Given a space of molecules of interest, find the smallest set of molecules that needs to be labeled (i.e., for which property values have to be obtained) such that a model trained on this set can be reliably applied to the rest of the space. Core set selection is valuable when the set of the molecules (or space of interest) is known a priori, such as in evaluating the properties of a molecule library or computing the properties of intermediates in a reaction network for further analysis. In previous work, we employed a selection strategy that balanced diversity-maximizing exploration of the space with the exploitation of the existing model to identify the cover set to train sparse linear additivity models using an ε-greedy strategy.[48] In this work, we develop an active learning methodology for the popular graph-based deep neural network model, SchNet, and apply this method through illustrative examples to build models for a variety of molecular properties. The novelty of this work is in employing a diversity-maximizing approach using machine learned features that tracks both the embedding of the underlying data and information about the property of interest. Through the examples, we show that: (1) not all molecules in a space are equally informative and that often a subset, even if randomly chosen, is sufficient to train reliable models and (2) our proposed method substantially outperforms random sampling in most cases. We further identify cases wherein our method does not outperform random sampling, and thereby identify the requirements for the distribution of the original molecular space so that our active learning can be cost-efficient.

2 Methods

2.1 Dataset

We benchmarked the active learning strategies on 4 different datasets: QM7 dataset (with 7102 molecules included),[49,50] PHYSPROP dataset[51] containing properties like boiling point (5434), melting point (8698), and LogP values (13402). QM7 is the subset of the GDB-13 database,[49] it contains small organic molecules which go up to seven (7) non-H atoms, and their atomization energies are calculated by density function theory. PHYSPROP dataset is a public dataset that contains over 41000 molecules with their structures, names, and physical properties designed for QSAR studies. It consists of 13 different physicochemical and environmental properties. Mansouri *et al*[52] further curated the data to remove $\sim$ 10% of the molecules that had data mismatches or were duplicates.[53]. In this work, we perform the active learning study on this curated PHYSPROP dataset of its three properties that have the most molecules available; boiling point, melting point, and LogP values.

2.2 SchNet deep learning model

We begin the discussion with the description of the machine learning model in this work. We apply SchNet, a graph convolution neural network (GCNN), to map the molecule structure with its property. It specifically maps a molecule into a feature representation as an array of values with a few convolutional layers, and further obtains its property values from the representation with several fully connected layers and nonlinear activation. The structure of SchNet is shown in Figure 1; the architecture consists of three parts: embedding, interaction, and prediction. A molecule's configuration, *i.e.*, its positions (3D coordinates) and atom charges, is fed as input. For each type of atom in the molecule, a numerical array is formed randomly to represent the atom as the starting point and its values are adapted while training. The matrix consisting of all the arrays is then fed into the interaction blocks as a cluster of graph convolutional layers. In these layers, a refined representation of the molecule is obtained by acquiring the interaction of each atom and its neighbors. For the atom in the molecule, the interaction is quantified with continuous-filter convolutions and radial basis functions of its distance and neighbor atoms. Further details are given in Schutt et al.[54]

After the convolution process, the molecule is represented by a refined matrix of fixed dimension $(N \times p)$, wherein N is the maximum of atom numbers for the molecule in the dataset and p is the dimension of the convolutional layer's output. Note that for a given molecule, this matrix captures its atom types, positions as well as the interaction between the atoms in a refined manner, thus we consider it, or its derivative could be used to represent the molecules numerically. Finally, following a typical deep learning process, two fully connected layers are applied to convert the matrix into a property value through linear summation:

$$y = wx + b \tag{1}$$

where w is the parameters in the layers and b is the bias introduced. These two fully connected layers have half the number of neurons of the convolutional layers and one neuron in the last layer respectively. Shifted softplus activation function is introduced on the first fully connected layer to introduce nonlinearity. In this work, we use the atom-wise summation of the first fully connected layer in our active learning strategy. Additionally, due to the small training set size for the active learning study to begin with, two dropout layers are added; one between the ouput layer from the interaction layer and another between the fully connected layer, which can randomly drop weights in the neurons and thereby reduce the chance of overfitting.

2.3 Active learning

Recently, Sener *et al*,[45] proposed a core-set approach to maximize the diversity of data points in the training set for building a convolutional neural network for image classification. The main idea of the core-set method is to minimize the active learning loss by reducing a core-set loss, which refers to the difference between the average loss over the labeled points and the average loss over the entire dataset. While the loss in the entire dataset is unknown

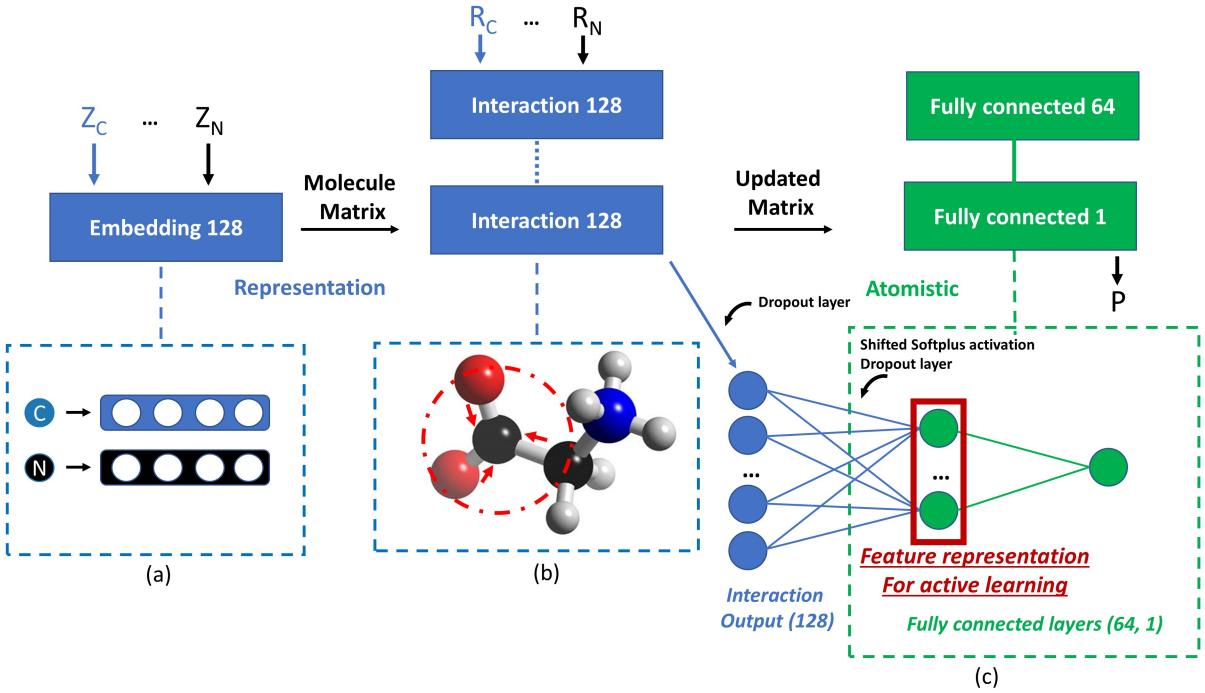

Fig. 1 The architecture of SchNet model to predict molecular properties. The configuration as atom charges and positions of the molecules is used as input, and goes through SchNet model to be converted into different properties. Additionally, one of the fully connected layers (in the brown box) is used to represent the molecule numerically and further applied for the active learning study. Specifically, ScheNet model consists of three parts; (a) the atom embedding layer wherein an numerical array initialized randomly for each atom type is assigned as the input for the neural network, (b) the multiple interaction layers with continuous filters to update information from neighbour atoms to the center atom; (c) two fully connected layers to map interaction output to property values.

(as not all labels are available), they show that in a classification problem using convolutional neural network, the loss is bounded by a value related to the number of classes, layers, and maximum sum of the weights in the layers; minimizing this upper bound would minimize active learning loss. Practically, they argue that active learning essentially reduces to finding the training set S (within a computation budget, b) to label such that the largest distance between each point in the unlabeled set (I) and its nearest point in S is minimized. Mathematically, this is:

$$\underset{S:|S|\leq b}{\text{Min}} \ \underset{i\in I}{\text{Max}} \ \underset{j\in S}{\text{Min}} \ \Delta_{i,j} \tag{2}$$

where $\Delta_{i,j}$ is a distance measure between two points i, j. This essentially is the K-center problem or a minimax facility location problem[55].

Such an algorithm has never been applied in the context of graph convolutional networks, particularly, for molecules. In this work, therefore, we adapt this algorithm for learning molecular properties and show its efficacy albeit without formal proof. The essential idea is to build a diversity-maximizing training set; this is intuitive because the more representative the subset is of the parent set, the better is the expected quality of trained models. Maximizing a diversity metric requires developing a way to represent the molecule and compute the distance between any pairs. Previously,[48] we used pathway fingerprints, essentially a set of atom traversal paths of different lengths on the molecular graphs, and then employed a simple Euclidean norm to compute the distance between molecules. However, such a fingerprint is handcrafted and does not keep track of the molecular property information. We, therefore, followed the approach of Sener *et al.*[45] and used the atom-wise summation of the first fully connected layer (see Figure 1) to represent the molecule; we argue that this vector captures the structural connectivity information of the molecule (thereby the functional group information) and also contains some information about the molecular property in question because of the first fully connected layer. This information, however, is not known *a priori*; therefore, the training set has to be constructed in an iterative manner. The first fully connected layer was identified to be the optimal choice based on an empirical study where we also considered other layers.

The proposed algorithm, to this end, is shown in Scheme 1. The starting point is the molecule space, U, under consideration; training budget n is the total number of molecules to add to the training set (which is often a good stopping criterion as it represents resource constraints for data acquisition) and batch size n_k is the number of molecules to add in each iteration. We typically set $n_k > 1$ to minimize the number of times the neural network has to be retrained. In the first step, n_0 molecules are randomly chosen from U and added to S. A neural network model (using, for instance, SchNet) is then used to train a model on S. Initially, the models are overparameterized, hence dropouts are particularly useful. Once the model, M, is trained for that iteration of the outer while loop, this information (see below) is used to it-

Algorithm 1 Dissimilarity-based selection

Input: molecule space U, training budget n, batch size n_k
Output: Training set S

1. Randomly select n_0 molecules from U and add to set S.
while $|S| \leq n$ **do**
2. Train neural network model, M, on S.
3. Extract features f from M.
4. Solve ***Max-min***(U, n_k, S, f) to get updated S.
end while
5. Return S.

Max-min(U, n_k, S, f):
$l = 0$
while $l < n_k$ **do**
d={}
for $i \in U - S$,
1. Compute $d_i = min(d_{ij}, i \in U - S, j \in S)$
2. Add d_i to d
Pick molecule $j = argmax(\{d\})$ from $U - S$ and add to S.
$l++$
end while

eratively add more molecules from $U - S$ into S in batches of n_k molecules, until the set S reaches the training budget or some other user-specified criterion is reached. The selection of the molecules is based on a diversity-maximizing criterion. In particular, a **max-min** problem is solved greedily, i.e., the shortest distance between each molecule in the remaining set $(U - S)$ and any molecule in the training set is computed and the molecule with the largest such distance is picked to add to the training set, the process then is repeated until the number of added molecules reaches n_k. The distance here is just the two-norm of the difference between the respective feature vectors based on M, i.e., $d_{ij} = |f_i - f_j|^2$) where f_i, f_j in the *Max-min* function are computed using the vector containing the atom-wise summation of the fully connected layer of the trained neural network model M applied to molecules i and j. The max-min is effectively equal to the method proposed by Wolf *et al.*[55] in finding the k-centers in U. As the model evolves, the features f that are employed change too.

We emphasize that our method does not explicitly compute prediction uncertainty to select new training data points, thereby avoiding the associated overhead computational costs. Nevertheless, since the distance in latent space can be used to compute uncertainties,[44], we posit that by sampling based on $d_{i,j}$, we may be implicitly considering some uncertainty information in our max-min selection. An appropriate end point for iterations (which we use here) is determined based on the computational budget (i.e., the maximum number of training data points 'n' to be used); however, one could also periodically compute uncertainty of the model (e.g. using a method that correlates with the latent space distances[44]) to determine the stopping point.

2.4 Implementation details

Three interaction blocks are used to map the interaction between atoms in the convolutional layers of SchNet models for all the properties and the number of the output neurons from the convolutional layers is set to be 128. The two fully connected subsequent layers have 64 neurons and 1 neuron respectively. Thus, the feature representation for a molecule is of the dimension of 1×64. (Note that the dimension of the refined matrix from the convolution process is $N \times p$, wherein N is the maximum the atom numbers, here dimension 1 is obtained with element-wise summation for each atom type in N.) During the training process, the epochs number for each iteration is set differently for each dataset. Boiling point and melting point models were trained for 400 epochs, while 250 epochs were sufficient for LogP and QM7. The number of randomly selected initial training set n_0 is set to be 400 for all the datasets, and the number of added points n_k for the batch are set to 160/320/400/200 for BP/MP/LogP/QM7 datasets respectively. The learning rate is fixed at 10^{-4}. Adam optimizer is used and mean absolute error (MAE) is set to be the loss metric.

3 Results and discussion

3.1 Active learning performance

We compare the performance of our active learning method with a random selection strategy at each iteration for the aforedescribed datasets. Since the data points are already labeled in these datasets, we use these studies purely as illustrative examples retrospectively to demonstrate and evaluate how our algorithm performs. Therefore, we use the property values of the molecules for training only when the molecule is moved into the set S. The error metric used is the mean absolute error (MAE) calculated on the remaining molecules (i.e. $U - S$) based on the version of the model at that iteration. It should be noted that the MAE can be calculated for these illustrative examples because the property values are available. Clearly, the two methods – active learning and random selection – will not result in the same training set, and therefore the remaining molecules are different as well. However, since our goal is to solve the core set problem (as described in the Introduction), our comparison of the error on the remaining molecules at a given training set size is appropriate. For a neural network, the small training sets often lead to overfitting, thus two dropout layer with the dropout rate set to be 0.2 is added.

Figure 2 shows this comparison for the four different properties. We conduct nine (9) different runs for the random method and three (3) for the active learning method. Then plot the average; the shaded region encompasses the minimum and maximum values of MAE at each iteration. We use a five-fold cross validation (CV) value as a benchmark for each of the plots; this represents an asymptotic limit of the model performance at nearly 80% of the space used for training. To fully understand the behavior of our proposed method, we simulated our active learning and random selection methods until one of them reaches this CV limit (as opposed to the maximum set size 'n' as described in the algorithm). We can observe that for most of the datasets, max-min based active learning significantly outperforms random sampling. Depending on the data set, the max-min method outperforms right from the start, or somewhere in the middle.

Two broad observations can be made from Figure 2: (1) in

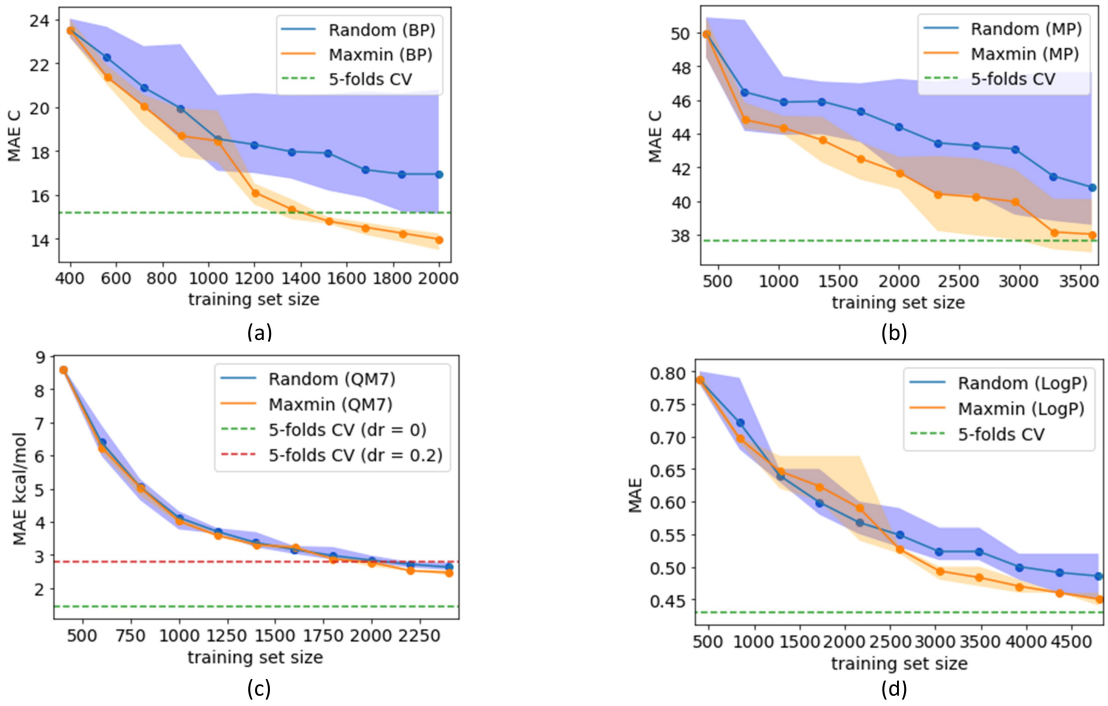

Fig. 2 The predicted mean absolute error(MAE) on the remaining set with the training set selected either by max-min method or randomly at each iteration. 4 different datasets are shown, where (a): Boiling point dataset, (b): Melting point dataset, (c): QM7 heat of atomization dataset, (d) LogP dataset. All models used two dropout layers, one between the ouput layer from the interaction layer and another between the fully connected layer. The dropout rate (dr) is set to be 0.2, while the second dashed line in (c) shows another condition where dr is set to 0.

all cases, the random sampling and active learning are both able to train models near the five-fold limit with a fraction of the data thereby indicating that not all molecules in the datasets are equally informative and (2) our proposed method outperforms random sampling for boiling point, melting point, and LogP while the performance of both methods was quite similar for the QM7 heat of atomization dataset. For the BP dataset, our method takes around 30% of the total data to reach the five-fold cross validation error limit while the MP and logP datasets require 40% and 35% respectively.

For the QM7 dataset, we can note that: (1) the max-min does not perform any better than the random method, (2) both methods reach the CV limit (for the model with a dropout rate of 0.2) at $\sim$ 30% of the data, and (3) the five-fold CV limit for the model with no dropouts is significantly better. Since SchNet was initially optimized for the QM9 dataset,[56] it is expected that reducing the dropouts increases the performance. For the other three cases, dropouts are valuable to prevent overfitting during early iterations.

3.2 Design application: identify top performing molecules

To illustrate the potential in molecule design, we applied our max-min strategy to the following problem: Given the space (library) of molecules, identify the top K highest performing molecules. In particular, we consider the BP, MP, QM7 heat of atomization, and LogP datasets where we compare the max-min and random sampling strategies to iteratively build neural network models and evaluate what fraction of predicted top 20 candidates (i.e. the top 20 highest BP, MP, heat of atomization, or LogP values) are in the true top 20 list. Figure 3 shows this match rate (in fraction) as a function of training set size for the two methods for all four datasets. We can observe that in the case of BP, MP, and LogP datasets, the match rate with active learning increases faster than with random sampling. For instance, even with one iteration, the match rate was 0.8 for the top 20 molecules of the BP dataset. In other words, the surrogate model trained on only 560 molecules is able to correctly identify 80% of the "top" molecules. Similarly, active learning correctly identifies 80% of the top 20 molecules in 9 and 6 iterations respectively for the MP and LogP datasets. Random sampling would have resulted in correctly identifying only about $35-70\%$ of the top molecules at the corresponding training set size.

In Figure 4, we compare the MAE of the predicted property of the true top 20 molecule for each of the four datasets at each iteration. Our method continues to outperform the random sampling for three datasets; BP, MP, and LogP. The MAE values for the top 20 molecules are significantly lower than for the random selection and the associated variance is smaller as well. The lower MAEs clearly explains the higher match rates; therefore, the results show the ability of the max-min strategy to identify top candidates under low computational budget. Finally, we note that the max-min strategy and random sampling have similar design performance (and MAE) for the QM7 dataset, consistent with the results shown in Figure 2.

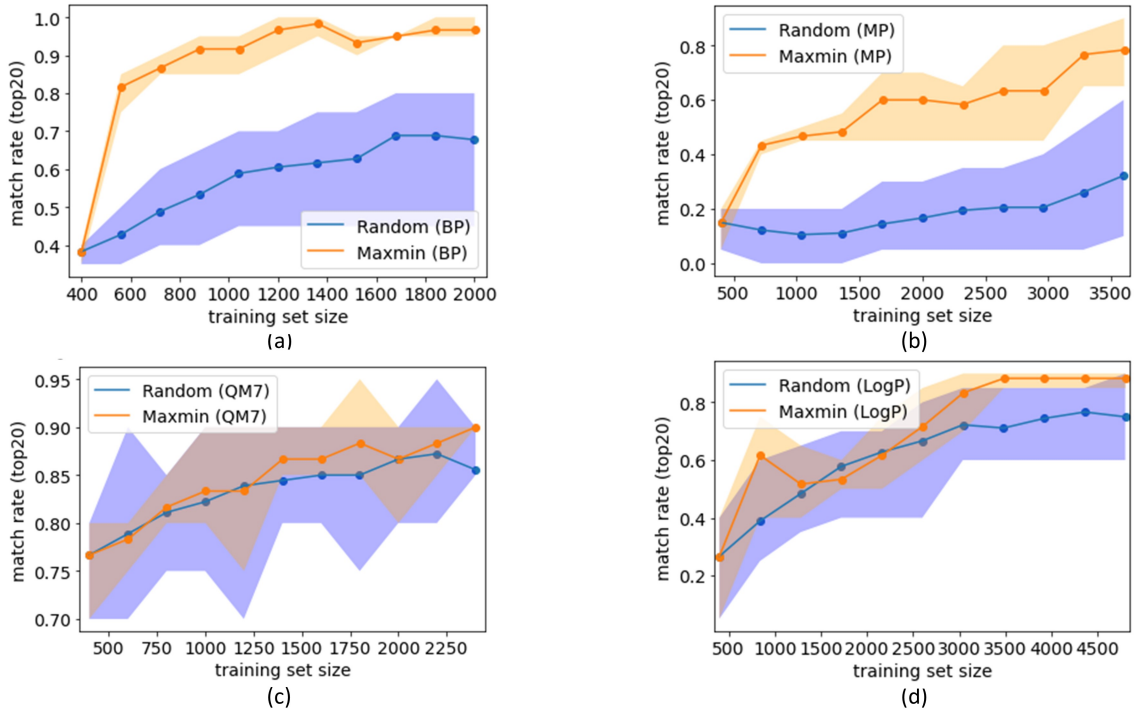

Fig. 3 An illustrative design problem to identify the top 20 molecules highest property values: Shown are the match rates at each iteration for the two sampling strategies for the four different datasets (a): Boiling point dataset, (b): Melting point dataset, (c): QM7 heat of atomization dataset, (d) LogP dataset. The match rate is the fraction of predicted top 20 molecules that are in the true top 20. The shaded regions capture the variance arising from the random initial set.

3.3 Analysis of the QM7 data vis-à-vis the boiling point

To understand why our algorithm did not offer any particular advantage over random sampling for QM7, we further investigated the two methods on QM7 and boiling point examples using data visualization and the distance metric. Figure 5 shows t-SNE plots to visualize the high-dimensional data distribution of the training set selected with the max-min and the random method for QM7; we also include the corresponding plots for training the boiling point model for comparison. The t-SNE method maps the high dimension features into a low dimension space such that the relative distance of the data points remains unchanged. For the sake of consistency, we use the feature function, f, generated from one of the CV models (trained on 80% of the data). We specifically visualize the training set after four iterations when the difference in MAE between the random and max-min methods for the boiling point model becomes noticeable in Figure 2. As we see in figures 5 (a,b), for the BP dataset, the distribution of the data for random sampling is different from that for the max-min method. In particular, active learning tends to sample the upper region of the plot (corresponding to higher values of the y-axis) more densely and the bottom region relatively sparsely. (Examples showing the molecules on different regions of the plot are included in Supporting information S1.) Additionally, we track the order of the batches of the added points, and color code them accordingly. Those plots and histograms showing all the iterations are included in Supporting information S2 & S3.

Figure 6 shows a comparison of the two methods based on the distance of each newly added point to the points already present in the training set for QM7 and the boiling point examples. This plot should indicate how dissimilar the added points are to those that are already in the training set as the active learning process proceeds. For the sake of consistency, we use the same distance metric as in the max-min algorithm but compute the feature vector using one of the five models developed for the five-fold cross validation. To reduce noise, we plot the average distance of every ten molecules added into the training set. For reference, we also included the mean pairwise distance between any two molecules in the complete set U for each case.

Since the max-min method tries to increase the diversity of the training set, we would naturally expect the distance of the newly added molecule in the max-min method to be generally larger than that for the random sampling (for the same training set size). Indeed, Figure 6(a) shows this trend for the boiling point example. The distances of molecules added in the initial stages of active learning are even larger than the mean distance of the complete space (and the difference in the distance of the added points between max-min and random sampling is comparable to this mean). This implies that the initial set (created via uniform sampling) did not adequately capture the complete space and the active learning algorithm tended to select dissimilar points (in a sense outlier points) from $U - S$ into the training set; with increasing training set size, the distance of the added points keeps decreasing, thereby pointing to the progressively greater diversity of the training set. For random sampling in the boiling point example, we can see that the distance of the added points is largely uniform regardless of the training set size indicating that the new

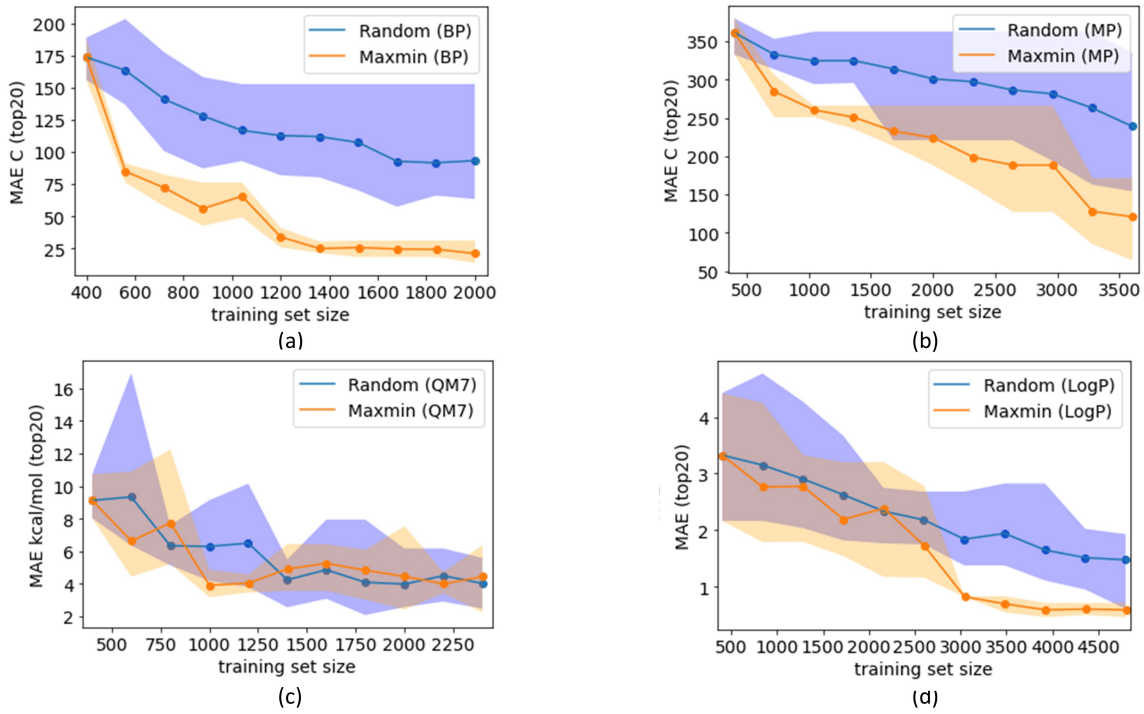

Fig. 4 The predicted mean absolute error(MAE) for the top 20 molecules with the largest property values in the dataset with the training set selected either by max-min method or randomly at each iteration. 4 different datasets are shown, where (a): Boiling point dataset, (b): Melting point dataset, (c): QM7 heat of atomization dataset, (d) LogP dataset. The shaded regions capture the variance arising from the random initial set.

points added are similar to the initial set (which was uniformly sampled). The two plots (orange and blue in Figure 6(a)) eventually overlap after more than 1000 points have been added.

For the QM7 example, the distances of the molecules added are much smaller than the mean pairwise distance of the complete set and the difference in the distance values between the two methods is relatively small compared to this mean. Since we start both methods with an initial uniformly sampled set of molecules, we can argue that the new points added are similar to the initial set. It should be noted here that QM7 is derived from the GDB dataset which in itself was curated through stochastic sampling techniques that maximized functional group diversity.[57] Therefore, that the random sampling behaved no differently from the max-min method indicates that the latent features learned by the neural network model for the heat of atomization are largely related to the functional group and structural features of the QM7 molecule space.

The boiling point data set is collated from experiments (coming from various sources) in contrast to how QM7 dataset was created; the differences in the behavior of the max-min strategy *vis-à-vis* random sampling for the two examples could arise due to the underlying functional group distribution of the two sets as well as the relative importance of capturing the outliers in the training set for the specific property. Nevertheless, the proposed max-min strategy reliably identifies smaller training sets with which reliable models can be developed.

4 Conclusion

Reducing the cost of training an accurate graph neural network model remains a challenge in computational chemistry. Here, we propose an algorithm based on the max-min method to cut down the budget for training an accurate deep learning model by actively constructing its training set. The algorithm aims to maximize the diversity in the training set represented by the latent features learned during the training process, which resembles a core-set approach problem. We demonstrate the effectiveness of the algorithm on the PHYSPORP dataset and QM7 dataset and show that it outperforms the uniform sampling in most cases; the algorithm succeeded in bringing down the budget of achieving a model of the same accuracy as using around 80% of the molecules selected uniformly in the space, while only use 30% $\sim$ 40% of the total data instead. For the QM7 dataset case wherein our algorithm does not outperform random sampling, we further visualize the dataset with the t-SNE method and compare that with the other case. We find out that in the visualization of the QM7 case the distribution of the points selected by our algorithm appears to be near-identical with the random sampling, while in the BP case our algorithm focuses on the edge of the dataset, and the clustered areas are better captured by the model. Also, the analysis based on the max-min metric shows that our algorithm brings in much more information in the BP case while failing in the QM7 case. Based on the findings, we conclude that our algorithm is suitable for the dataset which has a moderate functional group diversity, while could vary in size and structure.

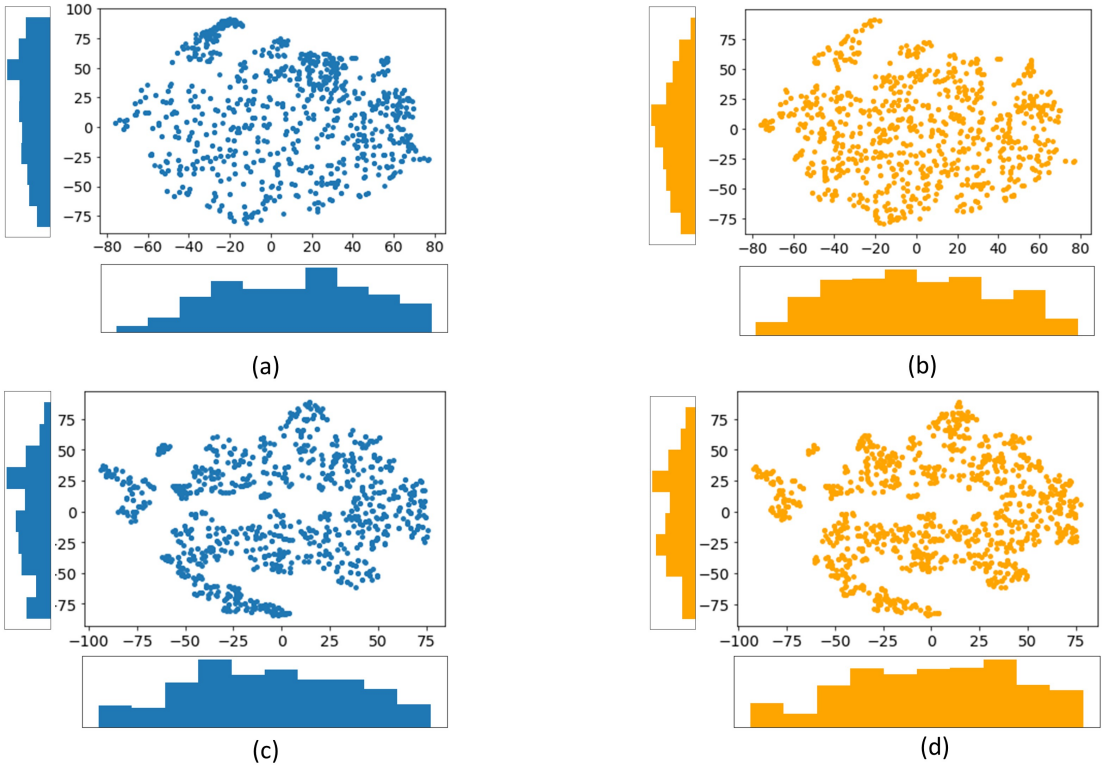

Fig. 5 The t-SNE visualization plot of the training set for the BP and QM7 examples using the max-min and random sampling after four iterations. (a) BP max-min, (b) BP random, (c) QM7 max-min, and (d) QM7 random. The distribution of the points selected by max-min or random sampling is also plotted as a histogram on the x and y axis.

Author Contributions

Bowen Li: Data curation, Methodology, Investigation, Software, Formal Analysis, Visualization, Writing – original draft.

Srinivas Rangarajan: Conceptualization, Methodology, Formal Analysis, Funding acquisition, Resources, Project administration, Supervision, Writing – review & editing.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

SR acknowledges partial financial support from Lehigh University and the National Science Foundation, CBET # 2045550. Portions of this research were conducted on Lehigh University's Research Computing infrastructure partially supported by the NSF Award 2019035.

Notes and references

1 C. Schober, K. Reuter and H. Oberhofer, *The journal of physical chemistry letters*, 2016, **7**, 3973–3977.

2 D. P. Tabor, L. M. Roch, S. K. Saikin, C. Kreisbeck, D. Sheberla, J. H. Montoya, S. Dwaraknath, M. Aykol, C. Ortiz, H. Tribukait *et al.*, *Nature Reviews Materials*, 2018, **3**, 5–20.

3 D. K. Dubey, D. Thakur, R. A. K. Yadav, M. Ram Nagar, T.-W. Liang, S. Ghosh and J.-H. Jou, *ACS Applied Materials & Interfaces*, 2021.

4 C. F. Perkinson, D. P. Tabor, M. Einzinger, D. Sheberla, H. Utzat, T.-A. Lin, D. N. Congreve, M. G. Bawendi, A. Aspuru-Guzik and M. A. Baldo, *The Journal of chemical physics*, 2019, **151**, 121102.

5 E. O. Pyzer-Knapp, C. Suh, R. Gómez-Bombarelli, J. Aguilera-Iparraguirre and A. Aspuru-Guzik, *Annual Review of Materials Research*, 2015, **45**, 195–216.

6 P. C. Rao and M. Yoon, *Energies*, 2020, **13**, 6040.

7 P. M. Modisha, C. N. Ouma, R. Garidzirai, P. Wasserscheid and D. Bessarabov, *Energy & fuels*, 2019, **33**, 2778–2796.

8 J. Lyu, S. Wang, T. E. Balius, I. Singh, A. Levit, Y. S. Moroz, M. J. O'Meara, T. Che, E. Algaa, K. Tolmachova *et al.*, *Nature*, 2019, **566**, 224–229.

9 E. Lionta, G. Spyrou, D. K Vassilatis and Z. Cournia, *Current topics in medicinal chemistry*, 2014, **14**, 1923–1938.

10 D. B. Kitchen, H. Decornez, J. R. Furr and J. Bajorath, *Nature reviews Drug discovery*, 2004, **3**, 935–949.

11 A. Stagni, Y. Song, L. A. Vandewalle, K. M. Van Geem, G. B. Marin, O. Herbinet, F. Battin-Leclerc and T. Faravelli, *Chemical Engineering Journal*, 2020, **385**, 123401.

12 H. Li, Z. Fang, R. L. Smith Jr and S. Yang, *Progress in Energy and Combustion Science*, 2016, **55**, 98–194.

13 S. Y. Foong, R. K. Liew, Y. Yang, Y. W. Cheng, P. N. Y. Yek, W. A. W. Mahari, X. Y. Lee, C. S. Han, D.-V. N. Vo, Q. Van Le *et al.*, *Chemical Engineering Journal*, 2020, **389**, 124401.

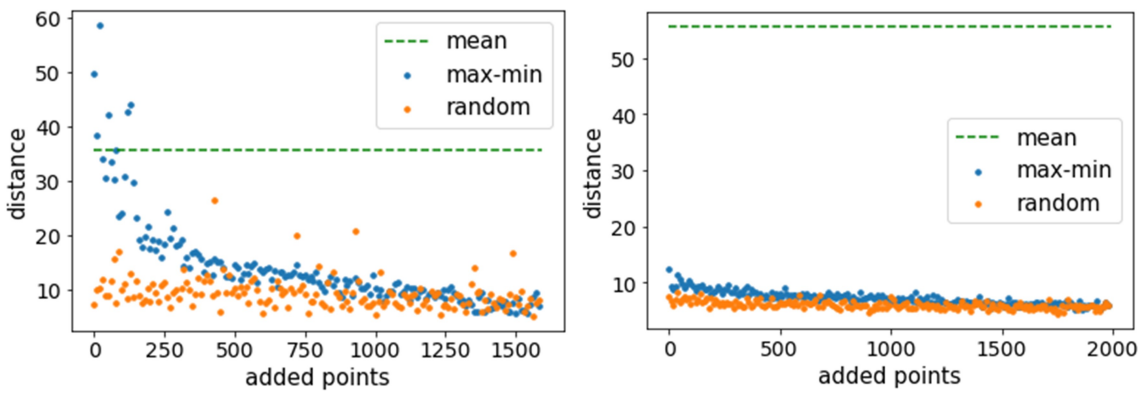

Fig. 6 The plot shows the minimum values of the distance between each added point and the training points during the active learning process. For illustration, the distance values of every 10 added points are averaged and shown in Figure. The dashed line shows the average pair-wise distance of the whole dataset. (a): BP dataset, (b) QM7 dataset.

14 A. S. Alshehri, R. Gani and F. You, *Computers & Chemical Engineering*, 2020, 107005.

15 D. C. Elton, Z. Boukouvalas, M. D. Fuge and P. W. Chung, *Molecular Systems Design & Engineering*, 2019, **4**, 828–849.

16 F. Gentile, V. Agrawal, M. Hsing, A.-T. Ton, F. Ban, U. Norinder, M. E. Gleave and A. Cherkasov, *ACS central science*, 2020, **6**, 939–949.

17 S. Korkmaz, *Journal of chemical information and modeling*, 2020, **60**, 4180–4190.

18 Q. Zhou, S. Lu, Y. Wu and J. Wang, *The journal of physical chemistry letters*, 2020, **11**, 3920–3927.

19 G. Subramanian, B. Ramsundar, V. Pande and R. A. Denny, *Journal of chemical information and modeling*, 2016, **56**, 1936–1949.

20 A. Aliper, S. Plis, A. Artemov, A. Ulloa, P. Mamoshina and A. Zhavoronkov, *Molecular pharmaceutics*, 2016, **13**, 2524–2530.

21 X. Ma, Z. Li, L. E. Achenie and H. Xin, *The journal of physical chemistry letters*, 2015, **6**, 3528–3533.

22 K. Vodrahalli, K. Li and J. Malik, *arXiv preprint arXiv:1811.12569*, 2018.

23 S.-J. Huang, J.-W. Zhao and Z.-Y. Liu, Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 1580–1588.

24 R. Liu and A. Wallqvist, *Journal of chemical information and modeling*, 2018, **59**, 181–189.

25 Y. Guo and D. Schuurmans, NIPS, 2007, pp. 593–600.

26 K. D. Konze, P. H. Bos, M. K. Dahlgren, K. Leswing, I. Tubert-Brohman, A. Bortolato, B. Robbason, R. Abel and S. Bhat, *Journal of chemical information and modeling*, 2019, **59**, 3782–3793.

27 D. Gissin and S. Shalev-Shwartz, *arXiv preprint arXiv:1907.06347*, 2019.

28 M. Ducoffe and F. Precioso, *arXiv preprint arXiv:1802.09841*, 2018.

29 C. Mayer and R. Timofte, Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 3071–3079.

30 R. Yuan, Z. Liu, P. V. Balachandran, D. Xue, Y. Zhou, X. Ding, J. Sun, D. Xue and T. Lookman, *Advanced materials*, 2018, **30**, 1702884.

31 Z. Del Rosario, M. Rupp, Y. Kim, E. Antono and J. Ling, *The Journal of Chemical Physics*, 2020, **153**, 024112.

32 K. Gubaev, E. V. Podryabinkin, G. L. Hart and A. V. Shapeev, *Computational Materials Science*, 2019, **156**, 148–156.

33 M. I. Zimmerman and G. R. Bowman, *Journal of chemical theory and computation*, 2015, **11**, 5747–5757.

34 L. Zhang, D.-Y. Lin, H. Wang, R. Car and E. Weinan, *Physical Review Materials*, 2019, **3**, 023804.

35 T. Young, T. Johnston-Wood, V. Deringer and F. Duarte, 2021.

36 J. S. Smith, B. Nebgen, N. Lubbers, O. Isayev and A. E. Roitberg, *The Journal of chemical physics*, 2018, **148**, 241733.

37 N. S. Eyke, W. H. Green and K. F. Jensen, *Reaction Chemistry & Engineering*, 2020, **5**, 1963–1972.

38 D. E. Graff, E. I. Shakhnovich and C. W. Coley, *Chemical Science*, 2021.

39 T. Lookman, P. V. Balachandran, D. Xue and R. Yuan, *npj Computational Materials*, 2019, **5**, 1–17.

40 A. A. Peterson, R. Christensen and A. Khorshidi, *Physical Chemistry Chemical Physics*, 2017, **19**, 10978–10985.

41 F. Musil, M. J. Willatt, M. A. Langovoy and M. Ceriotti, *Journal of chemical theory and computation*, 2019, **15**, 906–915.

42 K. Tran, W. Neiswanger, J. Yoon, Q. Zhang, E. Xing and Z. W. Ulissi, *Machine Learning: Science and Technology*, 2020, **1**, 025006.

43 L. Hirschfeld, K. Swanson, K. Yang, R. Barzilay and C. W. Coley, *Journal of Chemical Information and Modeling*, 2020, **60**, 3770–3780.

44 J. P. Janet, C. Duan, T. Yang, A. Nandy and H. J. Kulik, *Chemical science*, 2019, **10**, 7913–7922.

45 O. Sener and S. Savarese, *arXiv preprint arXiv:1708.00489*, 2017.

46 A. P. Soleimany, A. Amini, S. Goldman, D. Rus, S. N. Bhatia and C. W. Coley, *ACS central science*, 2021, **7**, 1356–1367.

47 W. Huang, D. Zhao, F. Sun, H. Liu and E. Chang, Twenty-fourth international joint conference on artificial intelligence, 2015.

48 B. Li and S. Rangarajan, *Molecular Systems Design & Engineering*, 2019, **4**, 1048–1057.

49 L. C. Blum and J.-L. Reymond, *J. Am. Chem. Soc.*, 2009, **131**, 8732.

50 M. Rupp, A. Tkatchenko, K.-R. Müller and O. A. von Lilienfeld, *Physical Review Letters*, 2012, **108**, 058301.

51 U. S. E. P. Agency, *US EPA(2014) EPI Suite data*, 2014, `https://www.epa.gov/tsca-screening-tools/epi-suitetm-estimation-program-interface`.

52 K. Mansouri, C. M. Grulke, R. S. Judson and A. J. Williams, *Journal of cheminformatics*, 2018, **10**, 1–19.

53 M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, K. Thiel and B. Wiswedel, *AcM SIGKDD explorations Newsletter*, 2009, **11**, 26–31.

54 K. Schutt, P. Kessel, M. Gastegger, K. Nicoli, A. Tkatchenko and K.-R. Muyller, *Journal of chemical theory and computation*, 2018, **15**, 448–455.

55 G. W. Wolf, *Facility location: concepts, models, algorithms and case studies. Series: Contributions to Management Science: edited by Zanjirani Farahani, Reza and Hekmatfar, Masoud, Heidelberg, Germany, Physica-Verlag, 2009, 549 pp. ISBN 978-3-7908-2150-5 (hardprint), 978-3-7908-2151-2 (electronic)*, 2011.

56 R. Ramakrishnan, P. O. Dral, M. Rupp and O. A. Von Lilienfeld, *Scientific data*, 2014, **1**, 1–7.

57 L. Ruddigkeit, R. Van Deursen, L. C. Blum and J.-L. Reymond, *Journal of chemical information and modeling*, 2012, **52**, 2864–2875.