Title

Graphical Gaussian Process Regression Model for Aqueous Solvation Free Energy Prediction of Organic Molecules in Redox Flow Battery

Authors

Gao, Peiyuan
Yang, Xiu
Tang, Yu-Hang
et al.

Publication Date

2021-06-15

Peer reviewed

Graphical Gaussian Process Regression Model for Aqueous Solvation Free Energy Prediction of Organic Molecules in Redox Flow Battery

Peiyuan Gao,[†] Xiu Yang,[*,‡] Yu-Hang Tang,[¶] Muqing Zheng,[‡] Amity Anderson,[†] Vijayakumar Murugesan,[*,†] Aaron Hollas,[†] and Wei Wang[*,†]

[†]*Pacific Northwest National Laboratory, Richland 99352, USA*

[‡]*Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA 18015, USA*

[¶]*Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA*

E-mail: xiy518@lehigh.edu; vijay@pnnl.gov; wei.wang@pnnl.gov

Abstract

The solvation free energy of organic molecules is a critical parameter in determining emergent properties such as solubility, liquid-phase equilibrium constants, and pKa and redox potentials in an organic redox flow battery. In this work, we present a machine learning (ML) model that can learn and predict the aqueous solvation free energy of an organic molecule using Gaussian process regression method based on a new molecular graph kernel. To investigate the performance of the ML model on electrostatic interaction, the nonpolar interaction contribution of solvent and the conformational entropy of solute in solvation free energy, three data sets with implicit or explicit water solvent models, and contribution of conformational entropy of solute are tested. We demonstrate that our ML model can predict the solvation free energy of molecules at

chemical accuracy with a mean absolute error of less than 1 kcal/mol for subsets of the QM9 dataset and the Freesolv database. To solve the general data scarcity problem for a graph-based ML model, we propose a dimension reduction algorithm based on the distance between molecular graphs, which can be used to examine the diversity of the molecular data set. It provides a promising way to build a minimum training set to improve prediction for certain test sets where the space of molecular structures is predetermined.

Introduction

Redox flow batteries (RFBs), particularly the aqueous organic RFBs (ORFBs), have gained significant interest for grid scale energy storage due to their inherent safety, flexible design, modular scale-up, and potential low cost. Critical functionalities of ORFBs such as energy density, cycling stability, and rate capability are largely impacted by the properties of the active organic species.[1,2] For example, the solubility of the active organic molecule dictates the energy density of an organic RFB. Therefore, the search for highly soluble (>1M) and chemically stable redox active organic materials has recently become a critical research endeavor.[3] The solubility, as well as the reactivity, viscosity, and redox potential of the active organic molecules depend on intricate interactions between the solute and solvent molecules, for which the free energy of solvation is often a critical parameter.[4,5] Evidently, solvation free energy has often been identified as a critical descriptor in quantitative structure-property/activity relationships (QSPR/QSAR) analysis. Yet there have been comparatively few experimental values (<2000) reported despite the millions of organic molecules synthesized to date. Density functional theory (DFT) and molecular dynamics (MD) simulation methods have been widely utilized for determining this prominent chemical descriptor.[6–12] With recent advancements in implicit solvation models[13–16] and operating functionals, the DFT and MD methodologies[17–20] provide a reliable estimate of solvation free energy with the mean-absolute-error approaching the chemical accuracy level of 1 kcal/mol. However,

approximations are often used to lower computational time at the cost of accuracy.[21,22] Furthermore, large-scale calculation of solvation free energy with high precision method through DFT and MD is computationally intractable. In view of this challenge, an artificial intelligence (AI) based prediction is needed because their computational strategies automatically improve through experience.[21,22] Machine learning (ML) methods are capable to predict a very broad range of properties. Recently, neural network model (NN) has received new attention for predicting solvation free energy prediction.[23–26] Some of these architectures operate over fixed molecular fingerprints common akin to traditional QSPR models.[27–29] However, due to the incomplete physical understanding of the structure of molecule and emergent properties, the features provided by domain experts may not include all critical design parameters in the material design. The graphical approach is a powerful tool to complement the domain experts knowledge because many features selected by domain experts are based on the computations which use the molecular structures.[30–35] Moreover, as molecules have arbitrary chemical composition and highly variable connectivity, useful information is difficult to be extracted from a molecule into a fixed dimensional representation. Thus, incorporating graphical approach can add important features that could be inadvertedly neglected by domain experts when designing an ML model. Naturally, a molecular structure can be represented by an undirected labeled graph that encodes both structural and functional information. The graph contains an initial feature vector and a neighbor list for each atom. The feature vector summarizes the atom's local chemical environment, including atom-types, hybridization types, and valence structures. Neighbor lists represent connectivity of the whole molecule. Another key question for molecular properties prediction using ML methods is lack of data, namely the data sparsity. Molecular properties data sets are different from the data sets in other applications as image recognition or natural language processing. Usually, the size of molecular properties data set that can be found is much smaller than those available for the aforementioned conventional machine learning tasks, as accurate results for molecular properties typically requires specialized instruments

and measurements. Therefore, the measurement cost of a small data set is rather expensive and time-consuming. Even for some molecular properties which can be obtained by computer simulation, e.g., solvation free energy in explicit solvent, the calculations are also not cost-effective. So the amount of training data remains a challenge in the property prediction of molecules.

Gaussian process (GP) is one of the most well studied stochastic processes in probability and statistics. Given the flexible form of data representation, GP is a powerful tool for classification and regression, and it is widely used in probabilistic scientific computing, engineering design, geostatistics, data assimilation, machine learning, etc.[36–38] In particular, given a data set comprising input/output pairs of locations and quantity of interest (QoI), GP regression (GPR), also known as Kriging, can provide a prediction along with a mean squared error (MSE) estimate of the QoI at any location. Alternatively, from the Bayesian perspective, GPR identifies a Gaussian random variable at any location with posterior mean (corresponding to the prediction) and variance (corresponding to the MSE). In other words, a GP model not only provides point predictions in the form of posterior means but also estimates the uncertainty of the prediction using posterior variances. Generally speaking, the larger the given data set size is, the closer the GPR's posterior mean is to the ground truth and the smaller the posterior variance is. While for small data set, the performance of GPR model is also good compared with deep neural network which typically requires a large training set.[39] Therefore, GP method is a good candidate for the machine leaning works when large data sets are difficult to be obtained.

In this work, we propose a machine learning model to predict the solvation free energy of organic molecules in water. We implement a graphical-kernel-based GP method[40,41] to construct surrogate models for solvation free energy prediction. In contrast to previous studies,[30–35] a weighted and labeled graph with labels on both nodes and edges in this work is used to give a more accurate representation for the inner structure of a molecule. Furthermore, to investigate the capability of our machine learning model on different components

of solvation free energy in thermodynamics as electrostatic interaction energy, the nonpolar interaction contribution of solvent and the contribution of conformational entropy of solute, we build and test three solvation free energy data sets, namely our own Pacific Northewest National Laboratory (PNNL) organic molecule data set, the QM9 data set, and the Freesolv data set. The solvation energy data in the three data sets include either the conformational entropy contribution or the effect of explicit solvent, or both of them. Our results are benchmarked against the three data sets. We demonstrate that our ML model can predict the solvation free energy of molecules at chemical accuracy (<1 kcal/mol) and 1000-10000 times faster than DFT/MD methods. Additionally, we try to elucidate the relationship between the molecular graph and molecular property using the model reduction method and provide a possible way on how to build a minimum training set to better predict the corresponding molecular property with ML model.

Method

GPR framework

We present a brief review of the GPR method adopted from Reference .[42,43] We denote the observation locations as $\boldsymbol{X} = \{\boldsymbol{x}^{(i)}\}_{i=1}^{N}$ ($\boldsymbol{x}^{(i)} \in D, D \subseteq \mathbb{R}^d$) and the observed values of the QoI at these locations as $\boldsymbol{y} = (y^{(1)}, y^{(2)}, \ldots, y^{(N)})^\top$ ($y^{(i)} \in \mathbb{R}$). For simplicity, we assume that $y^{(i)}$ are scalars. The GPR method aims to identify a GP $Y(\boldsymbol{x}, \omega) : D \times \Omega \to \mathbb{R}$ based on the input/output data set $\{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$, where Ω is the sample space of a probability triple. Here, $\boldsymbol{x}$ can be considered as parameters for this GP, such that $Y(\boldsymbol{x}, \cdot) : \Omega \to \mathbb{R}$ is a Gaussian random variable for any $\boldsymbol{x}$ in the set D. A GP $Y(\boldsymbol{x}, \omega)$ is usually denoted as

$$Y(\boldsymbol{x}) \sim \mathcal{GP}\left(\mu(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')\right), \tag{1}$$

where ω is not explicitly listed for brevity, $\mu(\cdot) : D \to \mathbb{R}$ and $k(\cdot,\cdot) : D \times D \to \mathbb{R}$ are the mean and covariance functions (also called *kernel* function), respectively:

$$\mu(\boldsymbol{x}) = \mathrm{E}\left\{Y(\boldsymbol{x})\right\}, \tag{2}$$

$$k(\boldsymbol{x},\boldsymbol{x}') = \mathrm{Cov}\left\{Y(\boldsymbol{x}), Y(\boldsymbol{x}')\right\} = \mathrm{E}\left\{(Y(\boldsymbol{x}) - \mu(\boldsymbol{x}))(Y(\boldsymbol{x}') - \mu(\boldsymbol{x}'))\right\}. \tag{3}$$

The variance of $Y(\boldsymbol{x})$ is $k(\boldsymbol{x},\boldsymbol{x})$, and its standard deviation is $\sigma(\boldsymbol{x}) = \sqrt{k(\boldsymbol{x},\boldsymbol{x})}$. The covariance matrix, denoted as $\boldsymbol{C}$, is defined as $C_{ij} = k(\boldsymbol{x}^{(i)},\boldsymbol{x}^{(j)})$. For any $\boldsymbol{x}^* \in D$, the GPR prediction and variance are

$$\hat{y}(\boldsymbol{x}^*) = \mu(\boldsymbol{x}^*) + \boldsymbol{c}(\boldsymbol{x}^*)^\top \boldsymbol{C}^{-1}(\boldsymbol{y} - \boldsymbol{\mu}), \tag{4}$$

$$\hat{s}^2(\boldsymbol{x}^*) = \sigma^2(\boldsymbol{x}^*) - \boldsymbol{c}(\boldsymbol{x}^*)^\top \boldsymbol{C}^{-1}\boldsymbol{c}(\boldsymbol{x}^*), \tag{5}$$

where $\boldsymbol{c}(\boldsymbol{x}^*)$ is a vector of covariance: $(\boldsymbol{c}(\boldsymbol{x}^*))_i = k(\boldsymbol{x}^{(i)},\boldsymbol{x}^*)$. Here $\hat{s}^2(\boldsymbol{x}^*)$ is also called the mean squared error (MSE) of the prediction because $\hat{s}^2(\boldsymbol{x}^*) = \mathrm{E}\left\{(\hat{y}(\boldsymbol{x}^*) - Y(\boldsymbol{x}^*))^2\right\}$.[43] Consequently, $\hat{s}(\boldsymbol{x}^*)$ is called the root mean squared error (RMSE).

In practice, it is common to assume that $\mu(\boldsymbol{x})$ is a constant function, i.e., $\mu(\boldsymbol{x}) \equiv \mu$. Also, the most widely used kernels in scientific computing are the Matérn functions, especially its two special cases, i.e., exponential and squared-exponential (Gaussian) kernels. For example, the Gaussian kernel can be written as $k(\boldsymbol{\tau}) = \sigma^2 \exp\left(-\frac{1}{2}\|\boldsymbol{x} - \boldsymbol{x}'\|_w^2\right)$, where the weighted norm is defined as $\|\boldsymbol{x} - \boldsymbol{x}'\|_w^2 = \sum_{i=1}^{d} \left(\frac{x_i - x_i'}{l_i}\right)^2$. Here, l_i $(i = 1, \ldots, d)$, the correlation lengths in the i direction, are constants. More details are provided in the support material.

Graph kernel

Using a graph kernel, the physical location $\boldsymbol{x}$ in the aforementioned conventional GP can take the form of a graph. In this work, we use each graph to represent a molecule. Therefore, each $\boldsymbol{x}$ can be considered as a molecule. We use the graph kernel to define notation of the

inner product between molecules and use it as the GP kernel $k(\boldsymbol{x}, \boldsymbol{x}')$. Following the notation in graph theory, we slightly modify the notation and use G to replace $\boldsymbol{x}$ in the GPR method. The practice of using labeled graphs, with the exemplary ball-and-stick model, to represent molecules gained popularity well before the era of machine learning.[44,45] In this work, we represent a molecule of n atoms as an undirected graph $G = \{V = \{v_i\}, E = \{e_{ij}\}, i, j \in \{1, \cdots, n\}$, where each atom i is represented by a vertices v_i that are labeled by chemical elements, charge, hybridization state, conjugacy, aromaticity, and hydrogen count.[46] Each edge $e_{ij} \in \mathbb{R}$ between vertices i and j represents the bond between between the atoms and is labeled by bond order, aromaticity, conjugacy, and ring membership. Its weight w_{ij} is set by a spatial adjacency rule $\mathcal{A}(\boldsymbol{r}_i, \boldsymbol{r}_j)$, which will be introduced later. Thus, the adjacency matrix $\boldsymbol{A}$ of a molecular graph is given as $A_{ij} = \mathcal{A}(\boldsymbol{r}_i, \boldsymbol{r}_j)$. Note that the edges are often supersets of the collection of covalent bonds in a molecule.

To implement the graph in a GP, we use the marginalized graph kernel $K(G, G')$,[40] which defines an inner product between two graphs, i.e., in our case, two molecules. The main idea is to perform random walks simultaneously on two given graphs and then calculate the expectation of the "similarity" between all pairs of the paths in such random walks. Specifically, each path, denoted as $\boldsymbol{h}$ on a graph, is the route from one atom to a certain one via chemical bonds in a molecule, and an inner product between the paths can be defined recursively using an element-wise inner products formula. Each $\boldsymbol{h}$ is a sequence consisting of vertices and edges:

$$v_{h_1} e_{h_1 h_2} v_{h_2} e_{h_2 h_3} v_{h_3} \cdots,$$

where v_{h_k} is the kth atom traversed by this path, and $e_{h_{k-1} h_k}$ is the chemical bond connection between the $(k-1)$th and the kth atoms in this path. Figure 1 shows an example of path between two nodes.

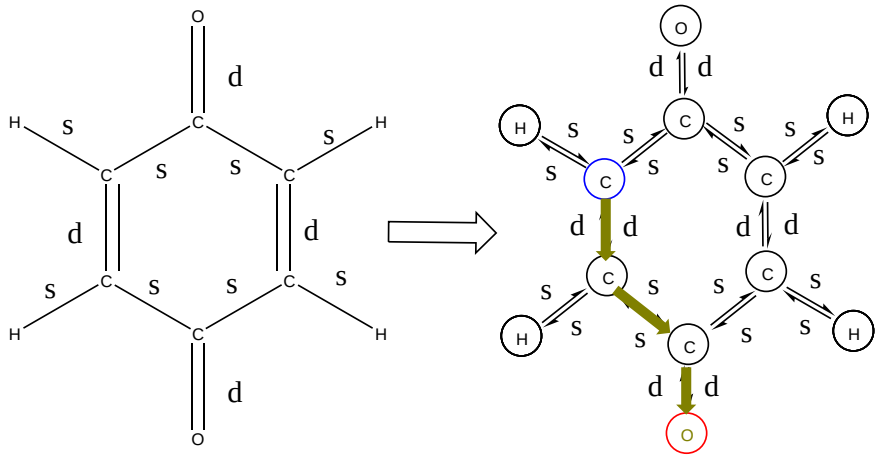


Figure 1: Demo of random walk on 1,4-benzoquinone molecule

The expectation of the path similarity in the simultaneous random walk is given as

$$K(G,G') = \sum_{\ell=1}^{\infty} \sum_{\boldsymbol{h}} \sum_{\boldsymbol{h}} \left(p_s(h_1) \prod_{i=2}^{\ell} p_t(h_i|h_{i-1}) p_q(h_\ell) \right) \times \left(p'_s(h'_1) \prod_{i=2}^{\ell} p'_t(h'_j|h'_{j-1}) p'_q(h'_l) \right) \times K_v(v_{h_1}, v'_{h'_1}) \prod_{k=2}^{\ell} K_v(v_{h_k}, v'_{h'_k}) K_e(e_{h_{k-1}h_k}, e'_{h'_{k-1}h'_k}). \tag{6}$$

Here, ℓ is the length of the path, $\boldsymbol{h}$ and $\boldsymbol{h}'$ are paths on the graphs represented by length-l vectors of vertex labels, ${}_s(\cdot)$ is the starting probability of the random walk on each vertex, $p_q(\cdot)$ is the stopping probability of the random walk on each vertex at any given step, $p_t(\cdot|\cdot)$ is the transition probability between a pair of vertices, $K_v(\cdot,\cdot)$ is a microkernel that computes the similarity between two vertices (i.e., atoms), and $K_e(\cdot,\cdot)$ is another microkernel that computes the similarity between pairs of edges (i.e., bonds).

Following the setup in,[41] we set the vertex elementary kernel as

$$K_v(v,v') = \begin{cases} 1, & \text{if} v = v' \\ \nu \in (0,1), & \text{otherwise.} \end{cases} \tag{7}$$

Here ν is a hyperparameter that will be learned using the training data set. The edge elementary kernel is a square exponential kernel (i.e., Gaussian kernel) function on edge

lengths, which is 1 if two edges are of the same length, and it smoothly changes to 0 as the difference in lengths grows:

$$K_e(e, e') = \exp\left[-\frac{1}{2}\frac{(e-e')^2}{\lambda^2}\right]. \tag{8}$$

The adjacency rule that computes the weights for each edge also assumes a square exponential form

$$\mathcal{A}(\boldsymbol{r}_i, \boldsymbol{r}_j) = \exp\left[-\frac{1}{2}\frac{\|\boldsymbol{r}_i - \boldsymbol{r}_j\|^2}{(\zeta\sigma_{ij})}\right] \tag{9}$$

where σ_{ij}, are element-wise length scale parameters derived from typical bonding lengths. A uniform starting probability $p_s(\cdot) \equiv s$ and a uniform stopping probability $p_q(\cdot) \equiv q$ are used across all vertices.

Given a training set D of m molecules and their associated solvation free energy $\{(M_1, \cdots, M_m)\}$, $\{(E_1, \cdots, E_m)\}$, as well as a marginalized graph kernel $K(\cdot, \cdot)$, the GPR prediction for the energy $\{E_1^*, \cdots, E_n^*\}$ of a test set of n unknown molecules $\{M_1^*, \cdots, M_n^*\}$ can be derived analytically as

$$\boldsymbol{E}^* := [E_1^*, \cdots, E_n^*]^\top = \boldsymbol{K}_{D^*}\boldsymbol{K}_{DD}^{-1}\boldsymbol{y}_D, \tag{10}$$

and the uncertainty in the prediction is given as:

$$\boldsymbol{\Sigma}^* := \boldsymbol{K}_{**} - \boldsymbol{K}_{D^*}^\top\boldsymbol{K}_{DD}^{-1}\boldsymbol{K}_{D^*}. \tag{11}$$

Here, K_{DD} is an $n \times n$ matrix with $K_DD(i, j) = K(M_i, M_j)$, $\boldsymbol{K}_{D^*}$ is an $n \times m$ matrix with $\boldsymbol{K}_{D^*}(i, j) = K(M_i, M_j^*)$ and $\boldsymbol{K}_{**}$ is an $m \times m$ matrix with $\boldsymbol{K}_{**}(i, j) = K(M_i^*, M_j^*)$.

Details of GPR

In the GPR method, the mean and covariance functions $\mu(\boldsymbol{x})$ and $k(\boldsymbol{x}, \boldsymbol{x}')$ are obtained by identifying their hyperparameters via maximizing the log marginal likelihood:[47]

$$\ln L = -\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^\top \boldsymbol{C}^{-1}(\boldsymbol{y}-\boldsymbol{\mu}) - \frac{1}{2}\ln|\boldsymbol{C}| - \frac{N}{2}\ln 2\pi. \tag{12}$$

Moreover, to account for the observation noise, one can assume that the noise is independent and identically distributed (i.i.d.) Gaussian random variables with zero mean and variance δ^2, and replace $\boldsymbol{C}$ with $\boldsymbol{C} + \delta^2\boldsymbol{I}$. In this study, we assume that observations $\boldsymbol{y}$ are noiseless. If $\boldsymbol{C}$ is not invertible or its condition number is very large, one can add a small regularization term $\alpha\boldsymbol{I}$ (α is a small positive real number) to $\boldsymbol{C}$, which is equivalent to assuming there is an observation noise. In addition, $\hat{s}$ can be used in global optimization, or in the greedy algorithm to identify locations of additional observations.

Given a stationary covariance function, the covariance matrix $\boldsymbol{C}$ can be written as $\boldsymbol{C} = \sigma^2\boldsymbol{\Psi}$, where $\Psi_{ij} = \exp(-\frac{1}{2}\|\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}\|_w^2)$. The estimators of μ and σ^2, denoted as $\hat{\mu}$ and $\hat{\sigma}^2$, are

$$\hat{\mu} = \frac{\mathbf{1}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{y}}{\mathbf{1}^\top\boldsymbol{\Psi}^{-1}\mathbf{1}}, \qquad \hat{\sigma}^2 = \frac{(\boldsymbol{y}-\mathbf{1}\hat{\mu})^\top\boldsymbol{\Psi}^{-1}(\boldsymbol{y}-\mathbf{1}\hat{\mu})}{N}, \tag{13}$$

where $\mathbf{1}$ is a constant vector consisting of 1s.[43] It is also common to set $\mu = 0$.[47] The hyperparameters σ and l_i are identified by maximizing the log marginal likelihood in Eq. (12). The terms $\hat{y}(\boldsymbol{x}^*)$ and $\hat{s}^2(\boldsymbol{x}^*)$ in Eq. (4) take the following form:

$$\hat{y}(\boldsymbol{x}^*) = \hat{\mu} + \boldsymbol{\psi}^\top\boldsymbol{\Psi}^{-1}(\boldsymbol{y}-\mathbf{1}\hat{\mu}), \tag{14}$$

$$\hat{s}^2(\boldsymbol{x}^*) = \hat{\sigma}^2\left(1 - \boldsymbol{\psi}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{\psi}\right), \tag{15}$$

where $\boldsymbol{\psi} = \boldsymbol{\psi}(\boldsymbol{x}^*)$ is a (column) vector consisting of correlations between the observed data and the prediction, i.e., $\psi_i = \frac{1}{\sigma^2}k(\boldsymbol{x}^{(i)}, \boldsymbol{x}^*)$.

Details of GPR using graph kernel

In Eq. (6), the straightforward enumeration is impossible, because ℓ spans from 1 to ∞. Nevertheless, Eq. (6) can be reformulated under the spirit of dynamic programming as follows:

$$K(G, G') = \sum_{h_1 \in V, h_1' \in V'} p_s(h_1) p_s'(h_1') K_v(h_1, h_1') R_\infty(h_1, h_1'), \tag{16}$$

where R_∞ is the solution to the following (linear) equilibrium equation

$$R_\infty(h_1, h_1') = p_q(h_1) p_q'(h_1') + \sum_{i \in V, j \in V'} t(i, j, h_1, h_1') R_\infty(i, j), \tag{17}$$

where

$$t(i, j, h_1, h_1') := p_t(i|h_1) p_t'(j|h_1') K_v(v_i, v_j) K_e(e_{ih_1}, e_{jh_1'}). \tag{18}$$

Equation 17 exhibits a Kronecker product structure, which can be readily recognized in matrix form:[41]

$$\mathbf{r}_\infty = \mathbf{q} \otimes \mathbf{q}' + \left[\left(\mathbf{P} \otimes \mathbf{P}' \right) \odot \left(\mathbf{E} \overset{\kappa_e}{\otimes} \mathbf{E}' \right) \right] \cdot \mathbf{diag} \left(\mathbf{v} \overset{\kappa_v}{\otimes} \mathbf{v}' \right) \cdot \mathbf{r}_\infty, \tag{19}$$

where

$\mathbf{v}$ is the vertex label vector of G with $\mathbf{v}_i = v_i$;

$\mathbf{p}$ is the starting probability vector of G with $\mathbf{p}_i = p_s(v_i)$;

$\mathbf{q}$ is the stopping probability vector of G with $\mathbf{q}_i = p_q(v_i)$;

$\mathbf{P}$ is the transition probability matrix of G defined as $\mathbf{D}^{-1}\mathbf{A}$;

$\mathbf{E}$ is the edge label matrix of G with $\mathbf{E}_{ij} = e_{ij}$;

$\mathbf{v}'$, $\mathbf{p}'$, $\mathbf{q}'$, $\mathbf{P}'$, $\mathbf{E}'$ are the corresponding vectors and matrices for G';

$\overset{\kappa_\mathrm{v}}{\otimes}$ is the generalized Kronecker product between $\mathbf{v}$ and $\mathbf{v}'$ with respect to microkernel κ_v;

$\overset{\kappa_\mathrm{e}}{\otimes}$ is the generalized Kronecker product between $\mathbf{E}$ and $\mathbf{E}'$ with respect to microkernel κ_e.

Machine learning model

Figure 2 presents a scheme of the predictive machine learning model framework by Gaussian process regression with graph kernel. First, the SMILES string of molecules in the data set are converted to graph, where the atoms are the nodes and the bonds are the edges. The graph kernel is then applied to average over the similarities of all paths generated from simultaneous random walks on each pair of graphs. A predictive model with Gaussian process regression can be built by the pairwise similarity matrix among the training molecules and the cross-similarity matrix between the new molecule and the training molecules.

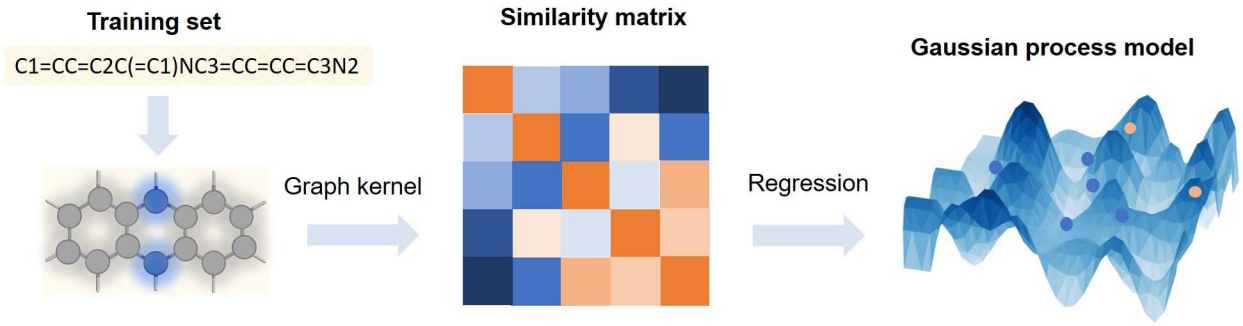


Figure 2: Scheme of the machine learning model pipeline

Metrics

In order to compare with the results, in this paper, mean absolute error (MAE) and root mean square error (RMSE) are applied to evaluate the performance of the ML model on the regression tasks.

$$\mathrm{MAE} = \frac{1}{n}\sum_{n=1}^{n} |\hat{y}_i - y_i|. \tag{20}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\Big(\hat{y}_i - y_i\Big)^2}. \tag{21}$$

where n is the number of molecules, y_i is the solvation free energy value in database, $\hat{y}_i$ is the prediction solvation free energy by the ML model.

Cross-Validation and Hyperparameter Optimization

We use the standard cross-validation approach to help identify the hyperparameters in the ML model, i.e., to perform model selection. For consistency, we maintain the same approach for all of our data sets. Specifically, for each data set, we split the data into training-validation and testing parts as described in Section . We employ 10-fold cross-validation (CV) for secure representation of the test data because the data set has a limited number of measurements. The molecules in the training-validation set of each data set is further split into 10 subsets following the sequence (InChIKey) of molecules. We choose one of the subsets as a validation set iteratively. The training set is the sum of the remaining 9 subsets. Consequentially, a 10-fold CV task performs 10 independent training and validation runs, and relative sizes of the training and validation sets are 9 to 1. We use Scikit-Learn library to implement the CV task and perform an extensive grid search for tuning hyperparameters. The hyperparameter set is determined by the result which has the minimum averaged MAE in the 10-fold CV. All the training is performed using our GPU-accelerated graph-kernel GPR tool.[41]

Results and Discussion

Database

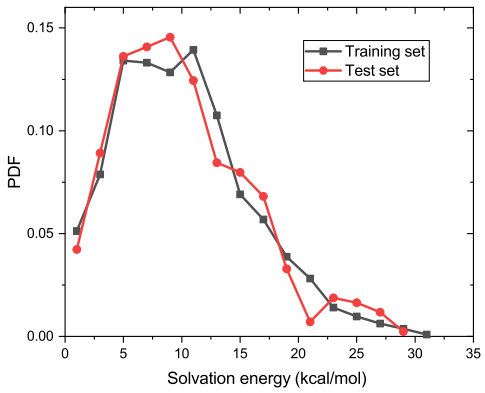
In order to test the performance of the model on the prediction of solvation free energy, three data sets are built. Data set A1 is the solvation energy data obtained from DFT

calculation with implicit water model. The molecules are selected from our own database. This solvation energy data set has 3626 molecules. All the molecules in the data set are neutral organic molecules. These molecules in the data set include ten types of elements, i.e., C, H, O, N, P, S, F, Cl, Br and I. All the solvation energy data in the data set are obtained from DFT calculation by PBE0 functional[48] at 6-31G** level[49] at 298.15K with NWChem code.[50] An effect of implicit water solvent with a dielectric constant of 78.4 is included via the COnductor like Screening MOdel for Real Solvents (COSMO) model.[16] These molecules are split into two sets as the training-validation set and test set following the sequence of their International Chemical Identifier key (InChIkey). Finally, 3200 molecules are selected in the training-validation set and 426 molecules are in the test set. Data set B1 is the solvation free energy data calculated by MD simulation in implicit water model. These data are obtained from a recently published paper.[51] The original molecules are chosen from the QM9 database. QM9 consists of 134k molecules with up to nine heavy atoms, including chemical elements C, H, O, N, and F. In this data set, molecules containing fluorine are removed by the authors. They randomly selected 4000 compounds from the QM9 database and calculated their solvation free energy by MD simulation with implicit water model. However, after carefully examining the InChikey of these molecules, we find 24 duplicates in the database. Therefore, we only select data from 3976 molecules from this database. Finally, 3600 molecules are used in the training-validation set and 376 molecules are in the test set. Data set C1 is obtained from the Freesolv database, which includes the solvation free energy both in experiment and MD simulation with explicit water model as solvent.[8] The experimental solvation free energy data are selected as our target in this work. To keep consistent with the other two databases, we do not use the solvation free energy data of chiral molecules in the Freesolv database. After excluding the chiral molecules, we select 588 molecules. The molecules in this database also include ten elements, i.e., C, H, O, N, P, S, F, Cl, Br and I. The 588 molecules are divided into two sets. The training-validation set includes 550 molecules and the test set has 38 molecules. Figure 1 shows the probability

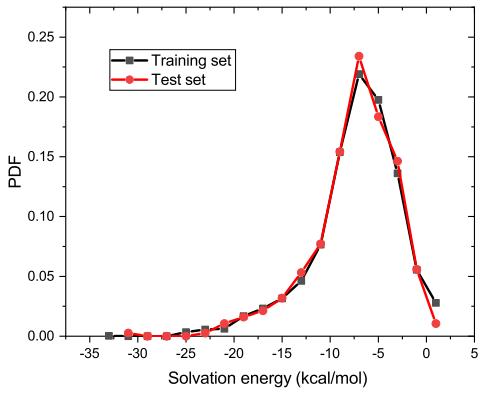
distribution function (PDF) of the training-validation set and test set for the three data sets. We can see that the train-validation set and test set in each data set have similar PDFs of solvation free energy. As the size of data set C1 is smaller, the fluctuation in the PDF is stronger than the other two databases. Overall, Figure 1 indicates that it is reasonable using the identifier InChIkey for random splitting data, especially when the data set is not very small, e.g., larger than one hundred molecules. In the ML model building, We use a Simplified Molecular Input Line Entry System (SMILES) string as initial input identifier in this work. The SMILES strings of molecules are converted to a graph with our graphic kernel when building ML models.

Solvation free energy prediction

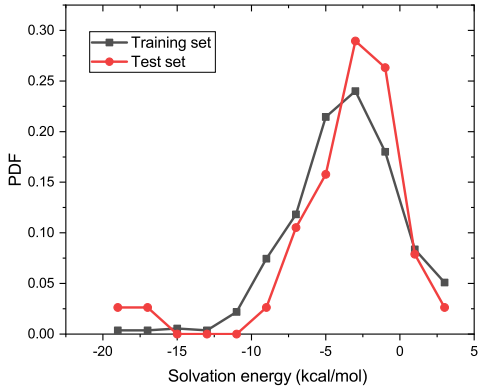
Solvation energies prediction results of the three data sets are displayed in Figure 4. With the help of optimized hyperparameters, the results of the three data sets show good performance for our ML model in general. The Pearson correlation coefficients R^2 between the truth and the prediction for the training set in the three data sets are 0.97, 0.98 and 0.95, respectively. The R^2 of the test set in these three cases are 0.91, 0.95 and 0.94, respectively. We can see the Pearson correlation coefficients are in good agreement for training data and test data in each data set, implying our ML model is not overfitted. The results in Figure 4 show that the predication accuracy for data sets B1 and C1 are better than for A1. The results are interesting, since in fact the measurement uncertainties of solvation free energy for the three data sets are increasing from A1 to C1. For DFT calculation, the measurement uncertainty for fixed functional and basis should be very small, as during the calculation the molecular conformation is fixed, and there is no thermal fluctuation. Therefore, the uncertainty should be $<$0.01 kcal/mol. In MD simulation with implicit solvent model, due to the conformational change in MD simulation, the fluctuation of calculated solvation free energy is larger than the DFT calculation, which increases measurement uncertainty. In experiments, the uncertainty can be even larger than the MD simulation, which has been



(a) A1



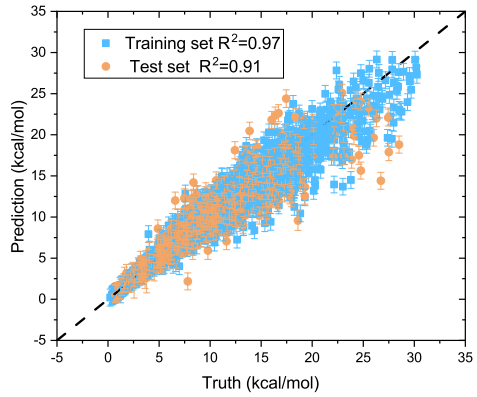
(b) B1



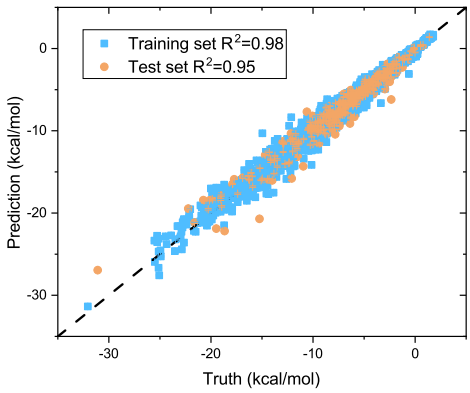
(c) C1

Figure 3: Probability distribution function of solvation free energy in training data set and test data set of the three data sets.(a) A1. (b) B1. (c) C1.

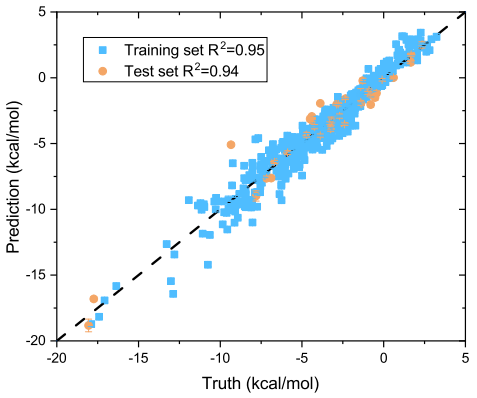
demonstrated in the Freesolv database. In the Freesolv database, the average error is about 0.06 kcal/mol for MD simulation data of solvation free energy, but for the experiment data it is 0.3 kcal/mol. However, by adding appropriate strength of white noise in the training process, we find that the uncertainty does not affect the accuracy of our ML models. Note that in general, it is necessary to include an appropriate level of measurement error, i.e., noise, to avoid overfitting when training ML models. In the GPR model, as indicated in Section , the noise is included in the covariance matrix. If the noise level included in the ML model is too small, the model is prone to overfitting. If it is too large, the error in prediction



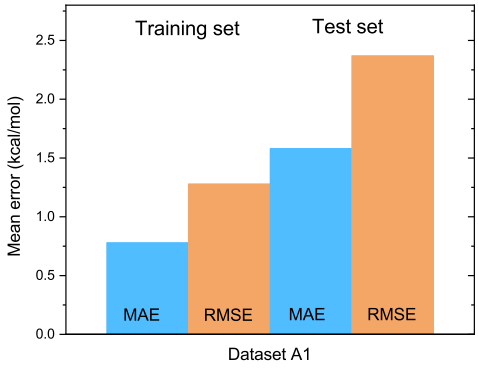
(a) Parity plot of data set A1



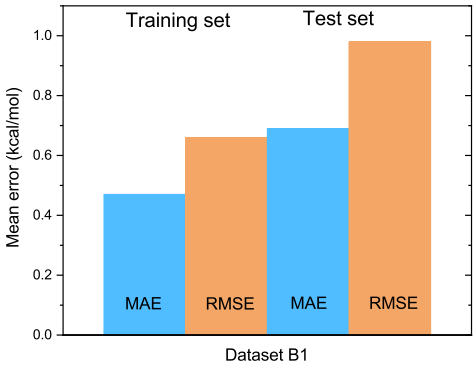
(b) Parity plot of data set B1



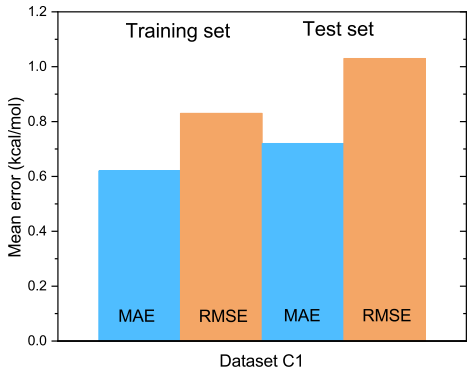
(c) Parity plot of data set C1



(d) MAE and RMSE in data sets A1



(e) MAE and RMSE in data sets B1



(f) MAE and RMSE in data sets C1

Figure 4: Parity plots, MAE and RMSE of training data and test data in data sets A1, B1 and C1.

would be also large. So noise is an important hyperparameter in the model parameterization.

Figure 4, parts d-f present the MAE and RMSE in training set and test set for the three data sets. For MAE results in both training set and test set in each data set, the results are very close, indicating our ML model is not overfitted. The RMSE results also show the same trend as MAE in each data set, which verifies our conclusion. For the training set in data set A1, the MAE is 0.78 kcal/mol and the RMSE is 1.28 kcal/mol. With regard to the test set in data set A1, the MAE and RMSE are close to the training set results but a little higher. The results are 1.58 kcal/mol and 2.37 kcal/mol, respectively. For the data set B1, the MAE and RMSE are 0.47 kcal/mol and 0.66 kcal/mol for training set. The test set follows the same trend. The MAE and RMSE are 0.69 kcal/mol and 0.98 kcal/mol. For data set C1, the MAE and RMSE result are close to the result obtained in data set B1. The MAE and RMSE in the training set are only a little higher than in B1. They are 0.62 kcal/mol and 0.83 kcal/mol. The test set results are similar, 0.72 kcal/mol and 1.03 kcal/mol, respectively.

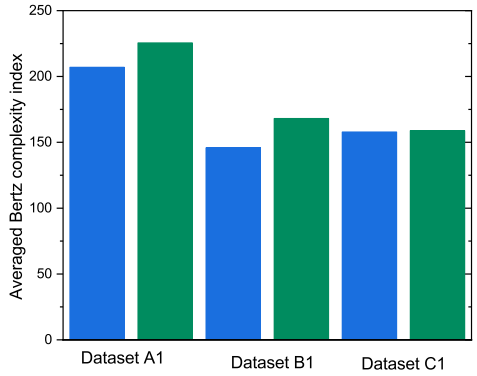
It is a bit difficult to directly compare our results with other ML models because we either have different data sets or use a different split method for the data set. While we know that the error of energy in a DFT calculation with different functional/basis would be several kilo calories, from the above results We can see that our ML model has yield chemical accuracy (1 kcal/mol) for the QM9 database subset and Freesolv database. Therefore, the mean absolute error in our ML model is actually close or even better than the DFT calculation. For the QM9 database subset, the authors previously obtained MAE = 0.7 kcal/mol with 2500 molecules in the training set,[51] while the MAE of our training set is 0.47 kcal/mol with 3600 training data. For Freesolv database, Wu et al. provided a benchmark study of 642 molecules with different QSPR/ML models.[52] The range of RMSE obtained with different ML methods is from 1.15 to 2.05 kcal/mol. In Lim and Jung's paper they obtained RMSE = 1.19 kcal/mol.[53] Our RMSE result is 1.03 kcal/mol with the same but even smaller training set. These results suggest that our graphic GP model guarantees considerably good performance.

The data set A1 has a large training set (3200 molecules), and theoretically the uncertainty of the data set A1 should be small. However, the performance of our model on data set A1 is not the best among the three data sets. For example, its R^2 is not the highest one of the data sets. One possible reason is that the complexity of this data set is higher. In data set A1 it involves ten types of elements. That means the converted molecular graph in data set A1 may have more types of nodes. In the view of graph theory, more types of nodes do not affect the topology, but they do increase the complexity of the molecular graph. Here, we use the Bertz complexity index to further characterize the complexity of the data set. The Bertz complexity index (BCI)[54] is defined as following

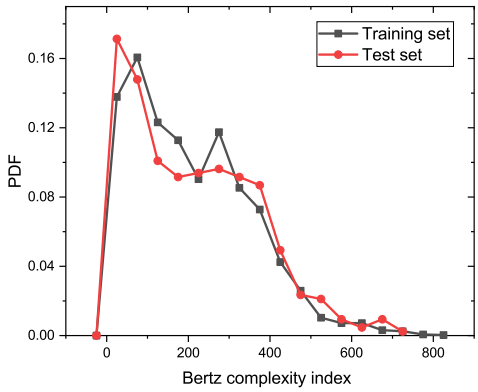
$$\mathrm{BCI} = 2n \log_2 n - \sum_l n_i \log_2 n_i, \tag{22}$$

where n is the number of pairs of adjacent edges in a graph G and n_i is the number of pairs of adjacent edges in the i-th class by symmetry. The term $n \log_2 n$ is used to prevent $\mathrm{BCI} = 0$ when all pairs of adjacent edges in G are equivalent. We can see that the first part takes into account structural characteristics of G, such as size, branching, and cyclicity, and the second part deals with the symmetry of G in terms of equivalent pairs of adjacent edges. In other words, one represents the complexity of the bonding, the other represents the complexity of the distribution of heteroatoms. BCI has been used in analysis of synthetic strategies in organic chemistry,[55] but it has not been connected to physical properties with the ML model. Figure 5a shows the average BCI values of the three data sets. It is found that the average BCI of the training set and the average BCI of test set in each data sets are very similar. The average BCIs obtained from training set and test set in data set A1 are 207.0 and 220.5, respectively. For the other two data sets the BCI values are 157.9 and 158.9 in data set B1, and 145.9 and 168.1 in data set C1 for training set and test set, respectively. The data set A1 has the largest BCI. It implies that on average, the converted molecular graph in data set A1 is the most complicated. Therefore, more training data may be needed

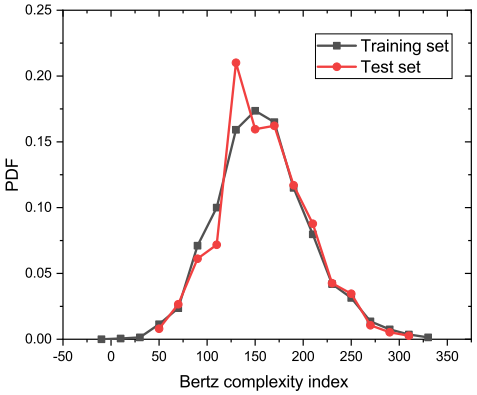
in order to reduce the MAE of the ML model on data set A1. The BCIs in data set B1 and C1 are close, although the type of elements in the two databases are not the same. It seems like the topological complexity in data set B1 and diversity of nodes in data set C1 have a complementary effect on BCI.



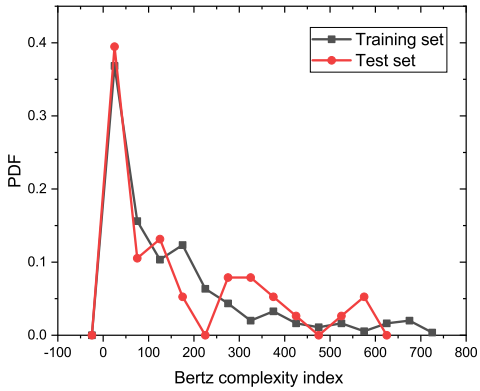
(a) The average BCI for each dataset



(b) The PDF of BCI for A1



(c) The PDF of BCI for B1



(d) The PDF of BCI for C1

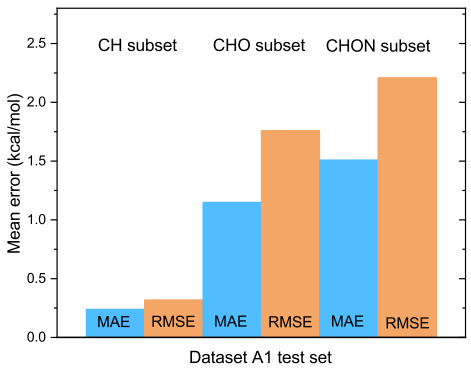
Figure 5: The average Bertz complexity index and PDFs of Bertz complexity index for datasets A1, B1, and C1. blue bar, training set. green bar, test set.

To further investigate the effect of BCI on performance of the ML model, we calculated the PDFs of BCI for each data set. Figure 5 parts b to d present the PDFs of BCIs in each data set. It reveals more details of the data sets. In all three data sets, the PDFs of BCI for training set and test set are very close, which is similar to the PDFs of solvation free energy.

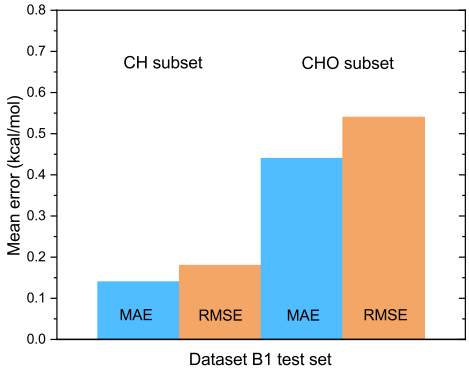
That validates the split method of data set with InChikey is effective again. In addition, we identify that the shape of the PDFs for data set A1 and C1 are similar. They are both long-tailed distributions, like a Poisson distribution. That may be because more types of elements are included in these two data sets, as they both have ten elements. The peaks of these two PDFs are both between 0 to 50, which means the small molecules are main components in BCI, but the contribution of large molecules to the average BCI cannot be neglected. In data set A1, the contribution of large or complicated molecules in the tail part is higher than data set C1. That makes the final BCI larger in data set A1 than data set C1. For data set B1, its distribution is close to a Gaussian distribution. It does not include more molecules with high BCI as in the other two data sets. Thus, eventually, the data sets B1 and C1 have similar averaged BCIs. Also, as shown above, the predictions of our ML model on these two data sets are consistent with their complexity. Based on these results, we can infer that for a complicated data set like the molecular data set, the performance of a graphic ML model is not only related to the absolute amount of training data, but also the data complexity. As the dimension of molecular data may be quite high, that infers the data sparsity problem in high dimensional space for training data.

For this reason, We do some tests with lower-dimensional subsets. We further evaluate the performance of our ML model with subsets in the test sets, which only include certain types of elements, e.g., C and H elements or C, H, and O elements. As shown in Figure 6, we see that all three data sets have the same trend. The MAE values increases with the element type complexity in these data sets. In these subsets, the simplest subset, which only includes the C and H elements, has the smallest MAE value. The MAE values are 0.24 kcal/mol, 0.14 kcal/mol, and 0.44 kcal/mol in data set A1, B1, and C1, respectively. These MAE values are much smaller than the MAE for the whole test set in these data sets. This is consistent with group contribution theory of solvation free energy, although the "groups" here are in high dimensional space. On the other hand, it indicates the ML model has relatively learned "more" information for compounds which only contain C and H elements from the training

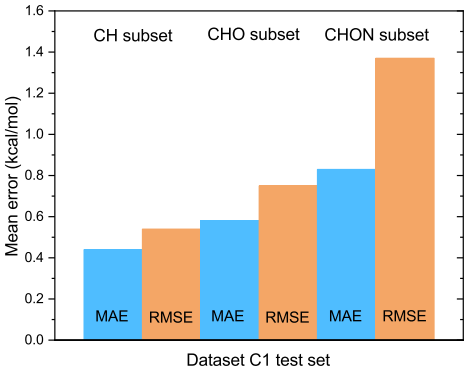
data. Additionally, we notice that the MAE value of the test group with C, H, O, and N elements in data set C1 is already higher than average in data set C1 test set (0.83 kcal/mol vs 0.72 kcal/mol), which implies the training data set is lacking molecules consisting of C, H, O, and N elements. The RMSE for the small test (1.37 kcal/mol) is also higher than the average value 1.24 kcal/mol.



(a) A1



(b) B1



(c) C1

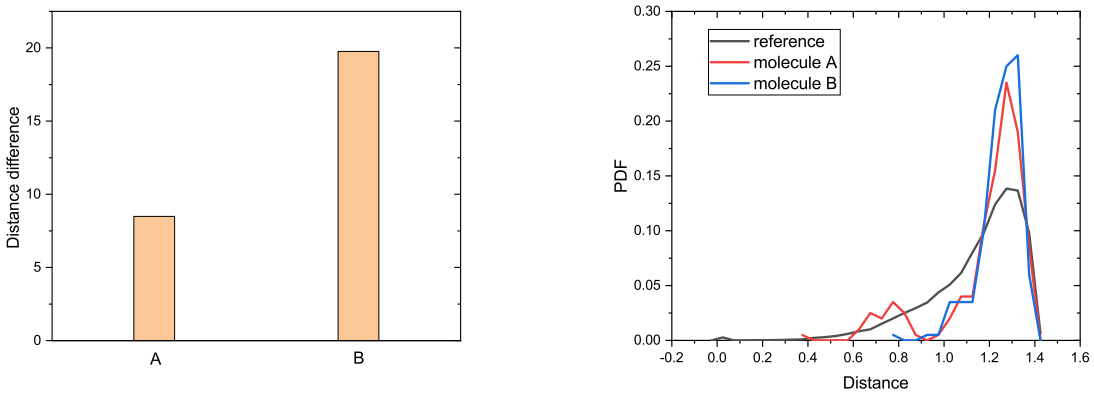
Figure 6: MAE and RMSE of different subsets in test data of data sets A1, B1 and C1. (a) A1. (b) B1. (c) C1.

Additionally, we provide a method to qualitatively estimate performance of the ML model on predicting properties of new molecules via comparing the distances between molecular graphs in the test set and training set. Here we show an example of a subset with 200

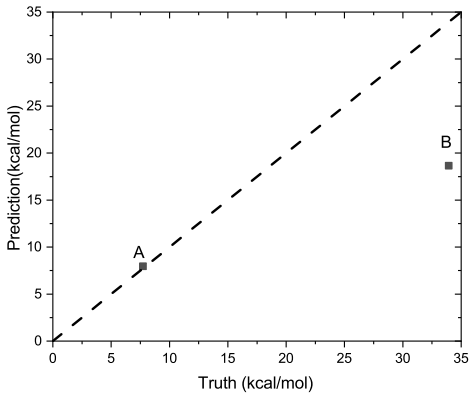
molecules in data set A1 and select two molecules as the illustrative test set. We calculate average pairwise distances between molecules in the training set, and between the training set and each test molecule. The average distances in training set and each test molecule are displayed in Figure 7(a). The PDFs of the distances are shown in Figure 7(b), which provides more details. We can find that the peak of PDF for molecule B is higher than molecule A, indicating the distance between the training set and B is farther than the distance between the training set and A in general. More importantly, the distances between molecule B and almost all training molecules are larger than 1.0, while there are some training molecules within the distance range of $[0.6, 0.8]$ from molecule A. Obviously, the distance for molecule A is much smaller than molecule B. In Figure 7(c) we can also see the solvation energy prediction of molecule A is much better than molecule B. An important reason is that there are a sufficient number of training molecules that are close to molecule A, which results in a prediction with greater accuracy.

Dimension reduction

To address the molecular data sparsity issue in high dimensional space and gain a deep understanding of the relationship between the training set and the ML model prediction, we analyze the training set with a model reduction approach. The covariance matrix that is used in the GP method plays a key role in the GPR, and it provides a possible way of exploring low-dimensional structures of the training data set that are critical to predict solvation free energy. In other words, it provides a possible way to identify critical functional groups (molecular fragments) that can be used as fundamental building blocks of real molecules, and the solvation free energy of a molecule can be predicted based on examining which groups are included in this molecule. To achieve this goal, we propose to associate molecules with points $Q_1, Q_2, \cdots, Q_m$ in Euclidean space $\mathbb{R}^d$, where d is the dimension to be identified. We aim to use the distance matrix of the aforementioned points in $\mathbb{R}^d$ to approximate the covariance matrix, as such to identify an appropriate d. This d is the number of the critical



(a) (b)



(c)

Figure 7: Distances, PDF and prediction of two example molecules A and B. (a) Average distances between the training set and test molecules A and B. (b) PDF of pairwise distances between training molecules, distances between the training molecules and molecules A, and B, respectively. (c) The actual number and prediction for solvation energy of molecule A and B with the ML model.

functional groups (or molecular fragments). Given a trained GP model and training data set, we have a covariance matrix $\boldsymbol{C}$. For a fixed d, we generate points in $\mathbb{R}^d$ based on this $\boldsymbol{C}$ as follows. We first define a matrix $\boldsymbol{T}$ as

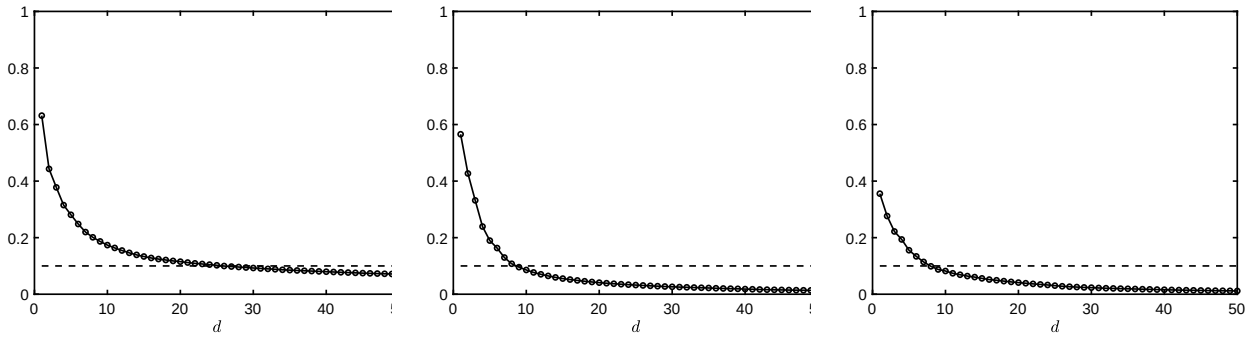
$$T_{ij} = \frac{C_{1j}^2 + C_{i1}^2 - C_{ij}^2}{2}. \tag{23}$$

Then we compute the eigenvalue decomposition of $\boldsymbol{T}$:

$$\boldsymbol{T} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^{\top}. \tag{24}$$

Finally, let $\boldsymbol{X} = \boldsymbol{U}\sqrt{\boldsymbol{S}}$, and the first d columns of $\boldsymbol{X}$ are the desired d-dimensional points in $\mathbb{R}^d$. Of note, the distance matrix of $Q_i, i = 1, 2, \cdots, m$ generated in this way, denoted as $\tilde{\boldsymbol{C}}$, is an approximation of the covariance matrix $\boldsymbol{C}$ when $d < m$. Although it is possible that $\tilde{\boldsymbol{C}} = \boldsymbol{C}$, we can set a threshold for the difference $\|\tilde{\boldsymbol{C}} - \boldsymbol{C}\|_F$ to examine the accuracy of the approximation. Here $\|\cdot\|_F$ is the Frobenius norm of a matrix.

Figure 8 illustrates the relative error $\|\tilde{\boldsymbol{C}} - \boldsymbol{C}\|_F / \|\boldsymbol{C}\|_F$ of the training data sets of A1, B1, and C1. In all cases, the relative error is smaller than 10%. This indicates that we only need to identify 8 critical functional groups to characterize the data sets B1 and C1 when predicting solvation free energy, which implies that these data sets have very good low dimension structure. We also notice that for data set A1, we need $d = 25$. This is consistent with the previous BCI analysis. As in data set A1, there are more types of elements (nodes). When we try to identify the critical functional groups/molecular fragments of the data set with model reduction approach, the effect of nodes (elements) on the number of critical groups is stronger than the topology of a molecule. Even though we do not have a strategy



(a) A1 (b) B1 (c) C1

Figure 8: Relative error $\|\tilde{\boldsymbol{C}} - \boldsymbol{C}\|_F / \|\boldsymbol{C}\|_F$ with respect to different d for different datasets. The dash line corresponds to 10% relative error. (a)A1. (b)B1. (c)C1.

to identify specific functional groups at the moment, the data analysis above shows potential

for achieving effective dimension reduction for molecules on solvation free energy prediction. We also note that, because the distance matrix of points in $\mathbb{R}^d$ is invariant under drift or rotation, identifying the map between basis in $\mathbb{R}^d$ and the critical functional groups requires comprehensive investigation and delicate design, which will be a target of our future work. In this work, we only show this potential via providing an abstract proof of concept in mathematics. This method is also valuable for predicting other properties.

Conclusion

In this work, we introduced a GPR model for solvation free energy prediction. The proposed GPR model used a marginalized graph kernel. A new similarity metric between molecules is defined in the marginalized graph kernel by both molecular topology and geometry. Therefore, the kernel can naturally adapt to molecules containing topological diversity and various types of elements. We benchmarked the performance of the GPR model on solvation free energy prediction across three data sets. To investigate the effect of different components in solvation free energy calculation as the effect solvent and contribution of conformational entropy, three solvation free energy data sets of our DFT calculations with implicit water model, a subset of QM9 database of MD simulation with implicit water model and a subset of experiment data in Freesolv database were built. We demonstrated that by tuning the hyperparameters, the uncertainty that was generated by explicit solvent and/or conformation change does not affect the accuracy of our GPR model. And we found that our GPR model with the marginalized graph kernel can predict solvation free energy at chemical accuracy ($<$1 kcal/mol) for the subsets of QM9 database and Freesolv database while using significantly small training data set (3% of QM9 database). Wu et al. have noticed that generally, the performance of graph-based model is better than other methods, but is not robust enough on complex tasks under data scarcity. We also identified the same issue for our ML model on data set A1. The complexity of these data sets were further analyzed

by model reduction method. We also found that the Bertz complexity index can be used to describe the data scarcity in high dimensional space to some extent. Finally, we showed a new method to evaluate the similarity between molecule in new test set and training set as well as the property prediction, which based on the distance between molecular graphs. This method provides a possible way on which to build a minimum training set to improve prediction for certain test sets. The current results show good performance of our GP model with graph kernel. Next step we will combine the current ML model with more descriptors to provide effective guidance for the inverse molecule design of organic molecules in a redox flow battery.

Data and Software Availability

All the training and test sets in this work are available with the paper (see the SI files). The GP model is build by scikit-learn library version 0.20.3 (https://scikit-learn.org/). The graph kernel is implemented by our GPU-accelerated python library Graphdot version 0.3.2 (https://github.com/yhtang/GraphDot). Additional data or code would be available upon reasonable request.

Acknowledgement

This work was supported by the Energy Storage Materials Initiative (ESMI), which is a Laboratory Directed Research and Development Project at Pacific Northwest National Laboratory (PNNL). PNNL is a multiprogram national laboratory operated for the U.S. Department of Energy (DOE) by Battelle Memorial Institute under Contract no. DE- AC05-76RL01830

Supporting Information Available

All the data sets for the machine learning model during this study are included in the Supplementary Information Files.

References

(1) Kwabi, D. G.; Ji, Y.; Aziz, M. J. Electrolyte Lifetime in Aqueous Organic Redox Flow Batteries: A Critical Review. *Chemical Reviews* **2020**, *120*, 6467–6489.

(2) Narayan, S. R.; Nirmalchandar, A.; Murali, A.; Yang, B.; Hoober-Burkhardt, L.; Krishnamoorthy, S.; Prakash, G. K. S. Next-generation aqueous flow battery chemistries. *Current Opinion in Electrochemistry* **2019**, *18*, 72–80.

(3) Gentil, S.; Reynard, D.; Girault, H. H. Aqueous organic and redox-mediated redox flow batteries: a review. *Current Opinion in Electrochemistry* **2020**, *21*, 7–13.

(4) Schnieders, M. J.; Baltrusaitis, J.; Shi, Y.; Chattree, G.; Zheng, L.; Yang, W.; Ren, P. The Structure, Thermodynamics, and Solubility of Organic Crystals from Simulation with a Polarizable Force Field. *Journal of Chemical Theory and Computation* **2012**, *8*, 1721–1736.

(5) Skyner, R. E.; McDonagh, J. L.; Groom, C. R.; van Mourik, T.; Mitchell, J. B. O. A review of methods for the calculation of solution free energies and the modelling of systems in solution. *Physical Chemistry Chemical Physics* **2015**, *17*, 6174–6191.

(6) Guthrie, J. P. A Blind Challenge for Computational Solvation Free Energies: Introduction and Overview. *The Journal of Physical Chemistry B* **2009**, *113*, 4501–4507.

(7) Tawa, G. J.; Martin, R. L.; Pratt, L. R.; Russo, T. V. Solvation Free Energy Calculations Using a Continuum Dielectric Model for the Solvent and Gradient-Corrected

Density Functional Theory for the Solute. *The Journal of Physical Chemistry* **1996**, *100*, 1515–1523.

(8) Duarte Ramos Matos, G.; Kyu, D. Y.; Loeffler, H. H.; Chodera, J. D.; Shirts, M. R.; Mobley, D. L. Approaches for Calculating Solvation Free Energies and Enthalpies Demonstrated with an Update of the FreeSolv Database. *Journal of Chemical & Engineering Data* **2017**, *62*, 1559–1569.

(9) Luukkonen, S.; Belloni, L.; Borgis, D.; Levesque, M. Predicting Hydration Free Energies of the FreeSolv Database of Drug-like Molecules with Molecular Density Functional Theory. *Journal of Chemical Information and Modeling* **2020**, *60*, 3558–3565.

(10) Subramanian, V.; Ratkova, E.; Palmer, D.; Engkvist, O.; Fedorov, M.; Llinas, A. Multisolvent Models for Solvation Free Energy Predictions Using 3D-RISM Hydration Thermodynamic Descriptors. *Journal of Chemical Information and Modeling* **2020**, *60*, 2977–2988.

(11) Jha, D.; Choudhary, K.; Tavazza, F.; Liao, W.-k.; Choudhary, A.; Campbell, C.; Agrawal, A. Enhancing materials property prediction by leveraging computational and experimental data using deep transfer learning. *Nature Communications* **2019**, *10*, 5316.

(12) Voityuk, A. A.; Vyboishchikov, S. F. Fast and accurate calculation of hydration energies of molecules and ions. *Physical Chemistry Chemical Physics* **2020**, *22*, 14591–14598.

(13) Cossi, M.; Rega, N.; Scalmani, G.; Barone, V. Energies, structures, and electronic properties of molecules in solution with the C-PCM solvation model. *Journal of Computational Chemistry* **2003**, *24*, 669–681.

(14) Tomasi, J.; Mennucci, B.; Cammi, R. Quantum Mechanical Continuum Solvation Models. *Chemical Reviews* **2005**, *105*, 2999–3094.

(15) Lin, S.-T.; Sandler, S. I. A Priori Phase Equilibrium Prediction from a Segment Contribution Solvation Model. *Industrial & Engineering Chemistry Research* **2002**, *41*, 899–913.

(16) Klamt, A. Conductor-like Screening Model for Real Solvents: A New Approach to the Quantitative Calculation of Solvation Phenomena. *The Journal of Physical Chemistry* **1995**, *99*, 2224–2235.

(17) Shivakumar, D.; Harder, E.; Damm, W.; Friesner, R. A.; Sherman, W. Improving the Prediction of Absolute Solvation Free Energies Using the Next Generation OPLS Force Field. *Journal of Chemical Theory and Computation* **2012**, *8*, 2553–2558.

(18) Kashefolgheta, S.; Oliveira, M. P.; Rieder, S. R.; Horta, B. A. C.; Acree, W. E.; Hünenberger, P. H. Evaluating Classical Force Fields against Experimental Cross-Solvation Free Energies. *Journal of Chemical Theory and Computation* **2020**, *16*, 7556–7580.

(19) Roos, K.; Wu, C.; Damm, W.; Reboul, M.; Stevenson, J. M.; Lu, C.; Dahlgren, M. K.; Mondal, S.; Chen, W.; Wang, L.; Abel, R.; Friesner, R. A.; Harder, E. D. OPLS3e: Extending Force Field Coverage for Drug-Like Small Molecules. *Journal of Chemical Theory and Computation* **2019**, *15*, 1863–1874.

(20) Fan, S.; Iorga, B. I.; Beckstein, O. Prediction of octanol-water partition coefficients for the SAMPL6-log P logP molecules using molecular dynamics simulations with OPLS-AA, AMBER and CHARMM force fields. *Journal of Computer-Aided Molecular Design* **2020**, *34*, 543–560.

(21) Fornari, R. P.; de Silva, P. Molecular modeling of organic redox-active battery materials. *WIREs Computational Molecular Science* **2020**, *n/a*, e1495.

(22) Sanchez-Lengeling, B.; Aspuru-Guzik, A. Inverse molecular design using machine learning: Generative models for matter engineering. *Science* **2018**, *361*, 360–365.

(23) LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444.

(24) Alshehri, A. S.; Gani, R.; You, F. Q. Deep learning and knowledge-based methods for computer-aided molecular design-toward a unified approach: State-of-the-art and future directions. *Computers & Chemical Engineering* **2020**, *141*, 19.

(25) Yang, J.; Knape, M. J.; Burkert, O.; Mazzini, V.; Jung, A.; Craig, V. S. J.; Miranda-Quintana, R. A.; Bluhmki, E.; Smiatek, J. Artificial neural networks for the prediction of solvation energies based on experimental and computational data. *Physical Chemistry Chemical Physics* **2020**, *22*, 24359–24364.

(26) Zubatyuk, R.; Smith, J. S.; Leszczynski, J.; Isayev, O. Accurate and transferable multitask prediction of chemical properties with an atoms-in-molecules neural network. *Science Advances* **2019**, *5*, eaav6490.

(27) Hutchinson, S. T.; Kobayashi, R. Solvent-Specific Featurization for Predicting Free Energies of Solvation through Machine Learning. *Journal of Chemical Information and Modeling* **2019**, *59*, 1338–1346.

(28) Riniker, S. Molecular Dynamics Fingerprints (MDFP): Machine Learning from MD Data To Predict Free-Energy Differences. *Journal of Chemical Information and Modeling* **2017**, *57*, 726–741.

(29) Ma, J.; Sheridan, R. P.; Liaw, A.; Dahl, G. E.; Svetnik, V. Deep Neural Nets as a Method for Quantitative Structure Activity Relationships. *Journal of Chemical Information and Modeling* **2015**, *55*, 263–274.

(30) Coley, C. W.; Barzilay, R.; Green, W. H.; Jaakkola, T. S.; Jensen, K. F. Convolutional Embedding of Attributed Molecular Graphs for Physical Property Prediction. *Journal of Chemical Information and Modeling* **2017**, *57*, 1757–1772.

(31) Kwon, Y.; Lee, D.; Choi, Y.-S.; Shin, K.; Kang, S. Compressed graph representation for scalable molecular graph generation. *Journal of Cheminformatics* **2020**, *12*, 58.

(32) Mahé, P.; Ueda, N.; Akutsu, T.; Perret, J.-L.; Vert, J.-P. Graph Kernels for Molecular Structure Activity Relationship Analysis with Support Vector Machines. *Journal of Chemical Information and Modeling* **2005**, *45*, 939–951.

(33) Mosbach, S.; Menon, A.; Farazi, F.; Krdzavac, N.; Zhou, X.; Akroyd, J.; Kraft, M. Multiscale Cross-Domain Thermochemical Knowledge-Graph. *Journal of Chemical Information and Modeling* **2020**, *60*, 6155–6166.

(34) Na, G. S.; Chang, H.; Kim, H. W. Machine-guided representation for accurate graph-based molecular machine learning. *Physical Chemistry Chemical Physics* **2020**, *22*, 18526–18535.

(35) Szczypiński, F. T.; Bennett, S.; Jelfs, K. E. Can we predict materials that can be synthesised? *Chemical Science* **2021**,

(36) Hu, X. S.; Xu, L.; Lin, X. K.; Pecht, M. Battery Lifetime Prognostics. *Joule* **2020**, *4*, 310–346.

(37) Lei, Y. G.; Li, N. P.; Guo, L.; Li, N. B.; Yan, T.; Lin, J. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing* **2018**, *104*, 799–834.

(38) Li, J.; Tartakovsky, A. M. Gaussian process regression and conditional polynomial chaos for parameter estimation. *Journal of Computational Physics* **2020**, *416*.

(39) Kamath, A.; Vargas-Hernandez, R. A.; Krems, R. V.; Carrington, T.; Manzhos, S. Neural networks vs Gaussian process regression for representing potential energy surfaces: A comparative study of fit quality and vibrational spectrum accuracy. *The Journal of Chemical Physics* **2018**, *148*, 241702.

(40) Kashima, H.; Tsuda, K.; Inokuchi, A. Marginalized kernels between labeled graphs. Proceedings of the 20th international conference on machine learning (ICML-03). 2003; pp 321–328.

(41) Tang, Y.-H.; de Jong, W. A. Prediction of atomization energy using graph kernel and active learning. *The Journal of chemical physics* **2019**, *150*, 044107.

(42) Abrahamsen, P. A review of Gaussian random fields and correlation functions. 1997.

(43) Forrester, A.; Keane, A.; Sòbester, A. *Engineering Design via Surrogate Modelling: A Practical Guide*; John Wiley & Sons, 2008.

(44) Tsuji, Y.; Estrada, E.; Movassagh, R.; Hoffmann, R. Quantum Interference, Graphs, Walks, and Polynomials. *Chemical Reviews* **2018**, *118*, 4887–4911.

(45) García-Domenech, R.; Gálvez, J.; de Julián-Ortiz, J. V.; Pogliani, L. Some New Trends in Chemical Graph Theory. *Chemical Reviews* **2008**, *108*, 1127–1169.

(46) Hamilton, W. L.; Ying, R.; Leskovec, J. Representation Learning on Graphs: Methods and Applications. *arxiv preprint* **2017**,

(47) Williams, C. K.; Rasmussen, C. E. *Gaussian processes for machine learning*; MIT press Cambridge, MA, 2006; Vol. 2.

(48) Perdew, J. P.; Ernzerhof, M.; Burke, K. Rationale for mixing exact exchange with density functional approximations. *The Journal of Chemical Physics* **1996**, *105*, 9982–9985.

(49) Ditchfield, R.; Hehre, W. J.; Pople, J. A. Self-Consistent Molecular-Orbital Methods. IX. An Extended Gaussian-Type Basis for Molecular-Orbital Studies of Organic Molecules. *The Journal of Chemical Physics* **1971**, *54*, 724–728.

(50) Aprà, E. et al. NWChem: Past, present, and future. *The Journal of Chemical Physics* **2020**, *152*, 184102.

(51) Rauer, C.; Bereau, T. Hydration free energies from kernel-based machine learning: Compound-database bias. *The Journal of Chemical Physics* **2020**, *153*, 014101.

(52) Wu, Z.; Ramsundar, B.; Feinberg, E.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. MoleculeNet: a benchmark for molecular machine learning. *Chemical Science* **2018**, *9*, 513–530.

(53) Lim, H.; Jung, Y. Delfos: deep learning model for prediction of solvation free energies in generic organic solvents. *Chemical Science* **2019**, *10*, 8306–8315.

(54) Bertz, S. H. The first general index of molecular complexity. *Journal of the American Chemical Society* **1981**, *103*, 3599–3601.

(55) Bertz, S. H. Convergence, molecular complexity, and synthetic analysis. *Journal of the American Chemical Society* **1982**, *104*, 5801–5803.