JCTC Journal of Chemical Theory and Computation

pubs.acs.org/JCTC Article

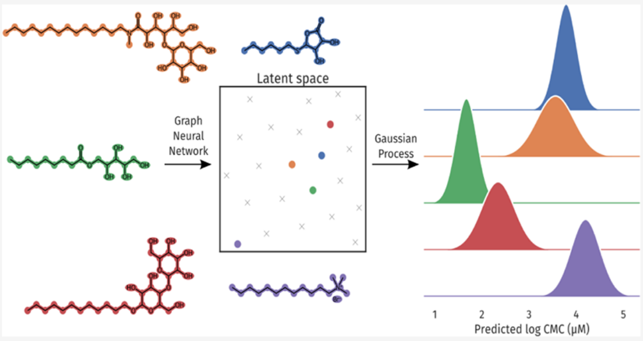
Analyzing the Accuracy of Critical Micelle Concentration Predictions Using Deep Learning

Alexander Moriarty, Takeshi Kobayashi, Matteo Salvalaglio, Panagiota Angeli, Alberto Striolo,* and Ian McRobbie

Cite This: J. Chem. Theory Comput. 2023, 19, 7371−7386 Read Online

ACCESS | Metrics & More | Article Recommendations

ABSTRACT: This paper presents a novel approach to predicting critical micelle concentrations (CMCs) by using graph neural networks (GNNs) augmented with Gaussian processes (GPs). The proposed model uses learned latent space representations of molecules to predict CMCs and estimate uncertainties. The performance of the model on a data set containing nonionic, cationic, anionic, and zwitterionic molecules is compared against a linear model that works with extended connectivity fingerprints (ECFPs). The GNN-based model performs slightly better than the linear ECFP model when there is enough well-balanced training data and achieves predictive accuracy that is comparable to published models that were evaluated on a smaller range of surfactant chemistries. We illustrate the applicability domain of our model using a molecular cartogram to visualize the latent space, which helps to identify molecules for which predictions are likely to be erroneous. In addition to accurately predicting CMCs for some surfactant classes, the proposed approach can provide valuable insights into the molecular properties that influence CMCs.



1. INTRODUCTION

The critical micelle concentration (CMC) of a surfactant defines the concentration above which the surfactant monomers self-assemble in solution to form micelles. The formation of micelles affects many interfacial phenomena.[1] However, predicting the CMC for an arbitrary chemical structure remains challenging.

Perhaps the most well-established predictor of CMC, X_{cmc}, is the Stauff–Kleven's relationship, first published in 1953.[2] It formalized the observation that CMC decreases exponentially with an increase in the number of carbons in the hydrocarbon tail, n_c

$$\log X_{cmc} = A - Bn_c, \; B > 0 \tag{1}$$

where A and B are empirical constants that depend on the temperature and the homologous series, i.e., the headgroup. The model is easily interpretable: to reduce CMC, it is sufficient to extend the surfactant's hydrocarbon tail, thus defining an easy-to-apply qualitative heuristic. Its drawback as a predictive model is its very limited applicability domain; each set of parameters is applicable only to surfactants with a specific headgroup and a linear carbon tail.

There has been a wealth of investigations into making more general models for CMC prediction; here, a brief review of some promising approaches will be given.

Puvvada and Blankschtein[3] derived a phenomenological model for studying aqueous nonionic surfactant systems that enabled the prediction of CMC and other properties across a range of temperatures. They developed a model for the free energy of micellization from which the CMC can be calculated. Their model was parametrized by three properties of the surfactant molecule:

- The tail length, defined as the number of carbon atoms.
- The average cross-sectional area of the headgroup, which controls the steric contribution to the free energy. This must be estimated.
- The Tolman length of the tail, which approximates the thickness of an "interaction region" around the tail.[4] This must also be estimated.

The authors developed a method to estimate parameters for linear, nonionic polyoxyethylene alkyl ether surfactants. The model attained a root-mean-squared error (RMSE) of

Received: August 8, 2023
Published: October 10, 2023

https://doi.org/10.1021/acs.jctc.3c00868

approximately 0.14 log μM for the group $C_{10}E_i$, where $i \in (3, 6)$, and 0.21 log μM for the group $C_{12}E_j$, where $j \in (3, 8)$. However, the error is much larger for other systems, like C_8E_6. The authors ascribed this inaccuracy to their prediction that the micelles in these systems would not exhibit much anisotropic growth; their model is better suited for cylindrical or disklike (bilayer) micelles.

The connection this model established between a small set of physically meaningful properties is extremely useful, especially because it does not explicitly require fitting to any experimental data. However, the procedures described for estimating the Tolman length are applicable only to linear hydrocarbon chains, not branched nor heterogeneous tail groups. Estimating the average cross-sectional area of the headgroup may also not be trivial.

Greater generalizability and better accuracy can be realized by considering the process of aggregation by identifying the behavior of individual atoms or small groups thereof. Simulation approaches can model the interaction of these units with each other and derive the potential energy of a configuration.[5−7] For example, molecular dynamic (MD) simulations treat individual atoms in the all-atom (AA) approach or groups of atoms in the coarse-grained (CG) approach as particles in a box that interact with each other. This allows the particles' movement to be simulated by integrating the equations of motion.

For example, Jorge[8] used an AA approach to simulate the self-assembly of *n*-decyltrimethylammonium bromide. They then estimated the CMC by considering the concentration of "free" surfactants, i.e., surfactants that were not in micelles, which they defined as an aggregate containing five or more surfactants. However, Jusufi and Panagiotopoulos[9] criticized the free surfactant concentration approach for modeling CMC of ionic surfactants in general. They note that the free surfactant concentration above the CMC is highly dependent on the choice of overall surfactant concentration, especially for ionic surfactants, which necessitates careful extrapolation to accurately determine CMCs.

Coarse graining, which groups atoms into beads, makes simulating longer time scales accessible.[10−13] For example, Khedr and Striolo[14] used dissipative particle dynamics (DPD) to model the CMCs of two polyoxyethylene octyl ether surfactants (C_8E_n). Rather than using the free surfactant concentration by itself, the authors calculated the volume fraction of the free surfactant in the accessible component of the aqueous phase. This approach accounts for the effective reduction in accessible volume due to the occupancy of micellar aggregates.[15] With this approach, they predicted CMCs for C_8E_1 and C_8E_9, both with errors of approximately 0.03 μM. Despite the high accuracy of the approach, there is a relative lack of validated DPD interaction parameters for different beads across a range of temperatures.[16]

Another approach is the conductor-like screening model (COSMO), which decomposes the problem by treating a molecule as a cavity with a charged surface in a solvent that acts as a dielectric continuum.[17] The cavity's surface is described by the solvent-accessible surface of the molecule. The geometry of this surface combined with a segmentwise description of its polarizing charges can be mapped using density functional theory (DFT).

COSMO for realistic solvation (COSMO-RS) adapts the model for more complex types of solvent.[18] Solvents only act like a dielectric continuum when they are capable of perfectly screening the COSMO-surface of a solute. COSMO-RS uses statistical mechanics to determine the probability distribution, describing how surface charge densities align between two molecules. This allows the chemical potentials to be determined.

Turchi et al.[19] used COSMO-RS to predict CMCs by treating a micelle as a separate phase and then considering the two-phase equilibrium between the micelle and an aqueous phase containing free surfactants. They modeled the micellar "phase" using two strategies. Their first strategy was to treat the micellar phase as being equivalent to a bulk, homogeneous phase of surfactant. The CMC was then determined by the equilibrium surfactant concentration in the aqueous phase. The authors argued that this approximation is more robust as the difference in polarity between the head and tail of a surfactant is reduced, which was the case for the majority of nonionic surfactants they considered.

Their second strategy was to consider the micellar phase as a bulk, homogeneous phase of an oil, whose chemistry was analogous to that of the surfactant's tail group. They then implemented an iterative procedure to calculate the interfacial tension (IFT) between the oil and aqueous surfactant phases at different concentrations of surfactant. The concentration at which the IFT is zero yields the CMC prediction. The premise of this approach is that the oil phase is representative of a micelle's interior, which is particularly true as the interactions between head and tail groups become more unfavorable. This is the case for highly polar headgroups: primarily for ionic surfactants.

The authors recommended applying both strategies and using the lower result as the CMC prediction. They attained an RMSE of 0.81 log μM on a data set of 24 surfactants containing a mix of ionic, nonionic, and zwitterionic surfactants. It is notable that the technique can be applied across all classes of surfactants.

Other approaches have extended COSMO-RS to explicitly account for the internal structure of micelles, such as COSMOmic, which treats a micelle as being made of concentric layers that each have their own surface charge profiles.[20] For example, Jakobtorweihen et al.[21] calculated CMCs using COSMOmic by first performing MD simulations to attain the layer-wise atomic distributions. The authors then predicted the CMCs of several polyoxyethylene alkyl ethers by determining the partition coefficients of inserting the respective surfactant monomer into a micelle.

COSMOplex is a recent extension of COSMOmic that removes the need to perform and initial MD simulation to determine the micellar structure.[22] Instead, it optimizes the micellar structure using a self-consistent approach, which iteratively yields new estimates for the layer-wise charge distributions. The authors predicted the CMCs of 10 nonionic surfactants with varied head and tail group chemistries, achieving an RMSE of 0.86 log μM.

Although COSMO techniques are promising, Herbert[23] note that many modern extensions are only available in the proprietary software package COSMOTHERM.[24]

Another approach to predict CMCs is to use an equation of state. For example, Li et al.[25] applied a segment-based UNIQUAC model (s-UNIQUAC) and a SAFT equation of state to predict CMCs of linear polyoxyethylene alkyl ethers by first deriving expressions for the activity coefficient of a surfactant in water.

Cheng and Chen[26] compared the performance of several models on a large data set: the polymer-NRTL model,[25] a UNIFAC model,[27] and a modified Aranovich and Donohue (m-AD) model.[26] The predictive performances of the models on the molecular series C_nE_6, C_nE_8, C_nE_9, $C_{10}E_n$, and $C_{12}E_n$ were compared, and the resulting RMSEs are summarized in Table 1. The models all have a reasonably good accuracy, but the SAFT model is particularly good.

Table 1. Comparison of the RMSEs of Selected Models On Polyoxyethylene Alkyl Ethers' CMCs[a]

model	RMSE ($\log \mu$M)
p-NRTL	0.18
s-UNIQUAC	0.14
SAFT	<u>0.06</u>
UNIFAC	0.14
m-AD	0.11

[a] The best RMSE is underlined. Data from Cheng and Chen.[26]

Segment-based semiempirical methods are very promising for predicting CMCs within a class of surfactants. Their major drawback is that they are applicable only to molecules that can be decomposed into segments that have trained parameters. In addition, they must respect the limitations of the theories on which they are based upon.

Finally, purely empirical methods have a heavy reliance on data abundance. Empirical methods offer a way of making predictions even when a unified theory is lacking or computationally too demanding. However, without an underlying theory, their limitations are not well-defined, and it is possible for the model to "learn" trends that contradict scientific intuition.

Empirical QSPR methods require validation to determine their reliability and applicability domain;[28−30] the performance metrics during optimization are not a reliable indicator of generality or the performance on new molecules. This is often achieved by partitioning the available data into *training* and *test* subsets; the former is used for optimizing the model's parameters, the latter is "hidden" from the model until training is complete, and the prediction metrics on the test data indicate how the model can be expected to perform in general. The test set should span the chemical space in which the model is intended to be applied.[30]

Empirical QSPR models can be used to design novel molecules with target properties[31,32] and are interpretable,[33] meaning that they can be analyzed to obtain chemical insights.

Mattei et al.[34] extended the Marrero and Gani group contribution method[35] to predict the CMCs of 150 nonionic surfactants. The descriptors are the number of groups present in a molecule. In the original method,[35] different "orders" of groups were identified; the first-order groups are forbidden from overlapping with one another and they are formulated so that any molecule of interest can be described using these groups exclusively. Higher-order groups distinguish polyfunctional molecules and isomers.[35]

Mattei et al.[34] introduced third-order groups to improve their model's accuracy by analyzing the molecules with the highest prediction errors after training an initial model with the first- and second-order groups from a prior work.[35] This is an example of *feature selection*, whereby the set of descriptors is expanded or contracted to adapt to the problem.[36,37]

The authors randomly selected 30 compounds as a test data set, achieving a RMSE of 0.13 log μM. The model is remarkably accurate and boasts high interpretability: the fitted contributions of each group describe their effect on the CMC quantitatively, and the existence of higher-order polyfunctional groups with large contributions implies that their constituent functional groups have a significant interaction with each other that affects the CMC. However, it may be difficult to determine whether a new molecule is within the applicability domain of the model, particularly because positional isomers are not necessarily distinguished from each other using the group representation.

Recently, an approach based on graph neural networks (GNNs) has produced highly accurate predictions while being applicable to nonionic, cationic, anionic, and zwitterionic surfactants.[38] Neural networks have many trainable parameters and a complex functional form. This ensures their versatility as universal approximators but makes them highly susceptible to overfitting.[39] Neural networks potentially boast the largest applicability domain (for a single set of trained parameters) of any model discussed previously.

GNN approaches operate on molecular graphs, which are characterized by atomic nodes whose edges represent bonds. Each operation on this graph considers just the local environment of an atom, i.e., the atoms that can be reached by traversing a single bond, but by stacking these operations in sequence, the size of the environment that is considered increases. In this sense, the model is similar to a group contribution approach, although the groups are determined by walking r steps along bonds from every atom in the molecule, where r is equal to the number of subsequent graph operations so that every group overlaps. Furthermore, each "contribution" is nonlinear, and the number of contributions is always equal to the number of atoms in a molecule. Because of their versatility, GNNs have been applied across a plethora of computational chemistry tasks from molecular property prediction[40−42] to enhanced sampling methods.[43,44]

Here, we build upon the previously published GNN model for CMC prediction[38] in two ways: we apply a hyperparameter search algorithm to further optimize the model's architecture and improve its accuracy, and we implement an *uncertainty quantification* technique that yields confidence intervals alongside CMC predictions. This improved model is compared against an adaptation of the group contribution approach that determines the entire set of groups to consider using a feature selection routine. In addition, a separate data set is introduced so as to perform external validation and to probe the limits of the model's applicability domain, as well as to analyze the uncertainty quantification.

Using the outcome of our approaches, we interpret both models and demonstrate a technique that allows one to visualize chemical space through the "eyes" of the trained GNN. By graphically mapping the molecular similarity, we demonstrate a new approach for creating visual, counterfactual explanations.[45] We show that this technique facilitates a better understanding of the applicability domain, which is essential to assess the accuracy of CMC predictions via neural networks and to build trust in the model.

2. METHOD

Two data sets were used for training and testing:

The Qin data set is a data set of 202 surfactants curated by Qin et al.[38] To the authors' knowledge, it is currently the largest public data set of CMCs for several classes of surfactants collected at standard conditions in an aqueous environment between 20 to 25 °C. In this work, the data were further subdivided into two tasks: Qin-All, the entirety of the data set, and Qin-Nonionics, which contains only the nonionic surfactants.

The Qin data were split into training and test subsets; the training data were used to fit the models, while test data were "locked away". The performance metrics on the test data were used for the evaluation of the models derived here. For some models, the training data were further split into optimization and validation subsets; the optimization data were used to calculate the loss function during model fitting. The validation data were used for on-the-fly evaluation of model performance during training to determine how many iterations of the optimization routine to use.

To provide a consistent benchmark of model performance with the previous work,[38] the same train/test data splits were used as Qin et al.[38] This will be termed the benchmarking test.

After training these models, a sensitivity analysis was carried out. The data for each task were split using a repeated, stratified k-fold cross-validation. Each molecule was assigned a class based upon its molecular fingerprint, which will be described below in the section on Extended connectivity fingerprints. The data were then split into k folds of roughly equal size and containing approximately the same percentage of molecules of each class, for $k \in (2, 5)$. For each fold, a model was trained using $k - 1$ folds and evaluated on the remaining fold. This was repeated thrice for $k = 2$ and twice for $k = 3$, each time with different randomization, and there were no repetitions for $k = 4$ and 5 so that there were a similar number of models trained for each value of k.

The Complementary data set contains 43 unique surfactant systems and their aqueous CMCs, extracted from the work of Mukerjee and Mysels.[46] The original document compiles CMC measurements for each system across several temperatures and with various additives. Each measurement was given a quality rating depending on the authors' assessment of the experimental method; measurements with a sufficiently good rating were categorized as "suggested" measurements. The data set used for this work was developed by taking the mean of the suggested CMC measurements between 20 to 25 °C, with no additives. The surfactant systems that contained elements that were not present in the data set (Mn, Cs, and Mg) were then pruned. The data set is provided in the Supporting Information.

The Complementary data set was used exclusively for external validation. Several of the surfactant systems in the data set are not expected to be within the applicability domain of the models derived here. This includes surfactants with counterions and combinations of functional groups that are not present in the training data. These data are included to test the robustness of the uncertainty quantification approach.

The number of each type of surfactant in the training and test subsets of the data is shown in Table 2. Only the models trained on the Qin-All data set were evaluated on the Complementary data set, as it contained ionic compounds.

The QSPR pipeline needs molecular descriptors and a functional form. The design processes for these are called *feature engineering* and *model selection*, respectively.

2.1. Feature Engineering.

In this work, two types of molecular descriptors are employed: extended connectivity fingerprints (ECFPs)[47] and molecular graphs.[48]

Table 2. Number of Each Type of Surfactant Contained in the Train/test Subsets of the CMC Data Sets[a]

data subset		number of			
task	train/ test	nonionics	anionics	cationics	zwitterionics
Qin-all	train	110	30	31	9
	test	12	4	4	2
Qin-nonionics	train	110			
	test	12			
complementary	train				
	test	12	23	6	2

[a] Missing entries mean no samples were in that class.

2.1.1. Extended Connectivity Fingerprints (ECFPs).

In the ECFP approach, the molecule is split into atomic environments up to a given radius, r: each environment is centered on an atom and extends r steps along connecting bonds. The set of all environments in the training data up to radius r is extracted. The resulting feature vector, $\vec{c}$, for a molecule has elements

$$c_m = \mathrm{Count}(\mathcal{E}_m) \tag{2}$$

where $\mathcal{E}_m$ represents the mth atomic environment.

A change in headgroup composition is reflected in a change in the subgraph counts. Provided that the new subgraph exists in the training data, the model can adjust its prediction accordingly. Branch points in a carbon chain are distinguished from main-chain groups, as they terminate in a CH group rather than CH_2.

ECFPs are similar to a segment-based approach; however, unlike segments or groups, subgraphs can overlap. While a group contribution approach requires that a canonical "priority" of the groups be defined prior to featuring molecules, by using ECFPs, the manual identification of important groups and their priorities are skipped; feature importance determination is delegated to the model.

Much like a group contribution approach, these fingerprints do not necessarily distinguish between all positional or chain isomers, particularly with smaller values of r, nor are stereoisomers treated differently.

It should be noted that, in this approach, the number of unique atomic environments is potentially very large relative to the size of the data available, which poses a risk of overfitting. Furthermore, larger environments necessarily envelop smaller ones, which means that there is some duplicate information in the representation: the presence of a $(CH_2)_3$ environment implies the presence of three CH_2 environments so that there is multicollinearity. This redundancy can impede the model fitting and interpretation. These issues can be ameliorated using a process of *feature selection*, which will be discussed below.

ECFPs are commonly employed for determining molecular similarity through the use of the Tanimoto similarity metric.[49−51] The Tanimoto similarity is a function of two binary fingerprints, so the count-based ECFPs are first converted by $b_i = \min(1, c_i)$. That is, if an atomic environment is present in a molecule, it is assigned a one; otherwise, it is assigned zero. The Tanimoto similarity between molecules A and B is then

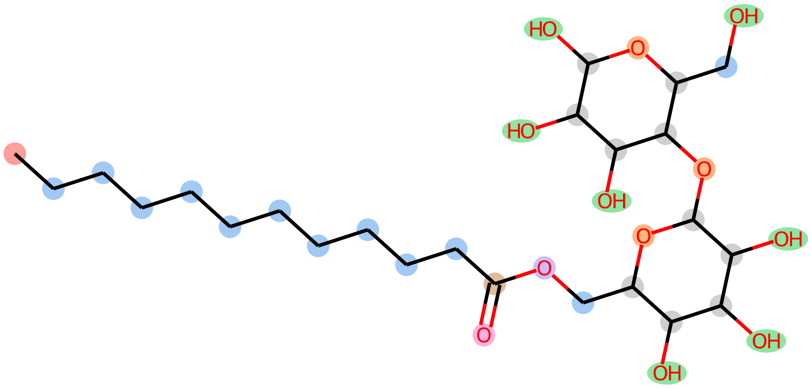


Figure 1. Molecular graph of 6-*O*-dodecanoyl-maltose. Atoms are highlighted based on their feature vectors, $\vec{v}_i$, so that equal feature vectors have the same color.

$$S_{AB} = \frac{\vec{b}_A \cdot \vec{b}_B}{\vec{b}_A \cdot \vec{b}_A + \vec{b}_B \cdot \vec{b}_B - \vec{b}_A \cdot \vec{b}_B} \tag{3}$$

In the original work by Tanimoto,[49] a "distance coefficient" is defined based on the logarithm of the similarity of two points; this is not a true distance metric as it does not satisfy the triangle inequality. Instead, the Jaccard distance, $d_{AB} = 1 - S_{AB}$, was employed here to perform clustering of the molecules of the Qin data set. The pairwise distances between each feature-selected molecular fingerprint in the Qin data set were computed, and the OPTICS clustering algorithm was used to assign a class to each of them.[52] The algorithm was parametrized with a minimum cluster size of 4 molecules. Outlying molecules, which did not fit into any other class, were assigned an "outliers" class for the sake of the stratified splitting procedure.

The clusters must be assigned before determining the train/test splits for each fold during the sensitivity analysis; therefore, in order to provide a canonical cluster assignment, the fingerprints resulting from the benchmarking test on the Qin-All task were employed.

2.1.2. Molecular Graphs. A molecular graph describes the entire topology of a molecule. Each atom is considered a *node* and each bond an *edge*. Rather than having a single feature vector to describe the molecule as a whole, each atom is assigned its own feature vector, $\vec{v}_i$, based on properties such as its element, hybridization state, charge, etc. The same set of atomic features was used here as in the work by Qin et al.[38] It is for this reason that the molecules in the Complementary data set containing elements that were not in the training data were excluded; the atomic representation includes a fixed-length one-hot encoding of the element number, and the new elements cannot be encoded without modifying the model after training. These feature vectors are concatenated into a node feature matrix, **V**. The graph's structure is then defined by a binary adjacency matrix, **A**

$$\mathbf{A}_{ij} = \begin{cases} 1 & \text{if } i \text{ bonded to } j, \text{ or } i = j \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Molecular graphs are natural representations to visualize; see Figure 1. This exact description of the molecule's topology enables an atomistic machine-learning approach.

2.2. Model Selection.

2.2.1. ECFP Model. Based on the prior knowledge encoded in eq 1, it is assumed that certain atomic environments have a linear relationship to log X_{cmc}. It therefore seems justified to apply a linear model to the ECFP fingerprints described in eq 2

$$\log X_{cmc} = \vec{w} \cdot \vec{c} + b \tag{5}$$

where $\vec{w}$ is a trained weights vector, the elements of which correspond to the contribution of an atomic environment to the CMC, and b is an intercept (or *bias* term).

However, the issues of the large feature vector size and multicollinearity must be addressed. To that end, a process of *feature selection* was applied, whereby a subset of the atomic environments was selected for use in the model. There are several approaches to feature selection;[36] here, we chose *regularization* due to the ease of implementation.

In this approach, we include a term in the loss function that depends on the norm of $\vec{w}$. The two types of constraints considered are l_1 and l_2 regularization, which correspond to the inclusion of l_1 and l_2 norms, respectively. The ElasticNet[53] loss function was employed

$$\min_{\vec{w}} \frac{1}{2n_{samples}} \left\| \mathbf{C}\vec{w} + \vec{b} - \vec{y} \right\|_2^2 + \alpha\rho \\ \|\vec{w}\|_1 + \frac{\alpha(1-\rho)}{2} \|\vec{w}\|_2^2 \tag{6}$$

where $n_{samples}$ is the number of training samples; $\vec{y}$ are the training data's true values of log X_{cmc}; $\vec{b}$ is a vector with elements all equal to b; and α and ρ are user-defined hyperparameters describing the degree of regularization ($\alpha \geq 0$) and the proportions of the regularization terms ($0 < \rho < 1$), respectively. **C** are the standardized training data feature vectors, $\{\vec{c}_n' \mid 1 \leq n \leq n_{samples}\}$, stacked rowwise into a matrix.

Standardizing the environment counts ensures that they have zero mean and unit variance

$$c_m' = \frac{c_m - u_m}{s_m} \tag{7}$$

where u_m and s_m are the mean and standard deviation of the number of $\mathcal{E}_m$ in each molecule in the training data. This standardization ensures that the regularization term is not dominated by environments with high variance and accounts for common and uncommon environments alike.

By imposing the l_1 penalty, the model is biased toward learning a *sparse* weight vector: many of its elements will be negligible. The corresponding features can be removed from the representation. Meanwhile, the l_2 penalty means that a "grouping" effect is achieved, ensuring that highly correlated

https://doi.org/10.1021/acs.jctc.3c00868

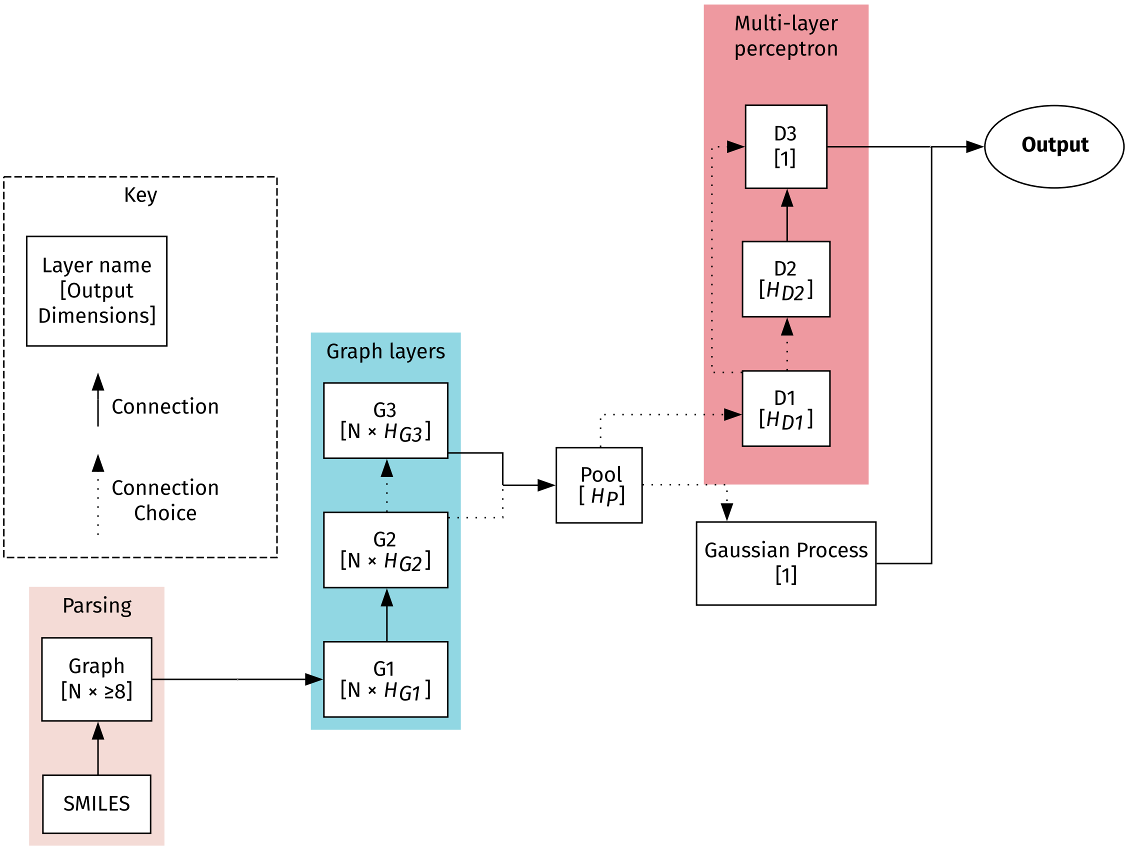


Figure 2. Schematic of the neural network architecture. Here, N represents the number of constituent atoms/ions in the input molecule and H represents a hyperparameter. The size of the pooling layer output, H_P, is independent only in the case of a gated attention-pooling layer. Otherwise, it is equal to the number of columns of the graph layer that feeds into it (H_{G2} or H_{G3}).

groups are assigned similar weights rather than discarding some of them. It also ensures that the upper bound on the number of selected groups is equal to the total number of groups; the model is not constrained to select a smaller subset if there are no redundant groups. Both of these are potential issues when using only an l_1 norm in the loss function.[53,54]

To determine the best values for α and ρ, 5-fold cross-validation of the training data was used. This was applied for a range of α and ρ combinations. The combination that achieved the lowest average mean-squared error was used to train a model by using the entirety of the training data. The hyperparameter search space is defined in the Supporting Information.

The features with non-negligible fitted weights from ElasticNet were then selected for use in the final linear model. This model, ridge regression, uses just l_2 regularization so that all of the weights are non-negligible but still addresses the issue of multicollinearity:[55]

$$\min_{\vec{w}} \left|\left|\mathbf{C}\vec{w} + \vec{b} - \vec{y}\right|\right|_2^2 + \alpha \left|\left|\vec{w}\right|\right|_2^2 \quad (8)$$

A similar cross-validation method to the one described above was used to determine the best α parameter, but using leave-one-out cross-validation, whereby $k = n_{\text{samples}} - 1$. Because only one hyperparameter needs to be determined, there are far fewer trials per fold, and therefore, a greater number of folds can be used.

It was empirically observed that the combination of ElasticNet feature selection and a final regression with the simpler ridge regression model yielded better results than the use of the weights from ElasticNet directly. This is likely due to using a larger number of folds when determining the best value for α. Both models were implemented using scikit-learn.[56]

2.2.2. Molecular Graph Model.

The basic topology of the graph neural network (GNN) was identical to the one used by Qin et al.[38] The first step of the model consists of a stack of graph network layers, which mutate the node features in a molecular graph based on those of bonded atoms. These layers employ the graph convolution network (GCN) architecture introduced by Kipf and Welling.[57] Layer l computes a new node feature matrix, $\mathbf{V}^{(l)}$, based on the adjacency matrix, $\mathbf{A}$:

$$\mathbf{V}^{(l)} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}\mathbf{V}^{(l-1)}\mathbf{W}^{(l)} + \mathbf{b}^{(l)} \quad (9)$$

Here, $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ are the weights and biases, respectively, of layer l. We have also introduced the degree matrix,

$$D_{ii} = \sum_j A_{ij} \quad (10)$$

so that the term $\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ normalizes the adjacency matrix based on the degree of each atom.

$\mathbf{V}^{(1)}$, therefore, encodes information about not only the atom itself but also its bonded neighbors. This information is used in the subsequent graph convolution so that $\mathbf{V}^{(2)}$ encodes information about the second-order neighborhood, *et cetera*. The number of graph layers, L, therefore dictates the "radius" around each atom that is considered in computing the final feature vector, analogous to creating an ECFP, except that the

i^{th} atomic environment is characterized by a continuous, *latent* vector, $\vec{v}_i^{(L)}$.

The next step is a pooling layer, which converts the graph to a single *latent representation vector*, $\vec{v}^{(p)}$, losing explicit topological information. Several choices of pooling function were trialed:

Mean pooling was employed by Qin et al.[38] It computes the average over all of the atoms' latent feature vectors.

Sum pooling computes the sum of all atoms' latent feature vectors. This is the most analogous to the ECFPs in that the contribution of an atomic environment scales linearly with the number of times it occurs in the molecule.

Gated attention pooling applies an *attention* mechanism to decide which environments are relevant to the prediction:[58]

$$\vec{v}^{(p)} = \sum_i^N \sigma(\mathbf{W}_1\vec{v}_i^{(L)} + \vec{b}_1) \odot (\mathbf{W}_2\vec{v}_i^{(L)} + \vec{b}_2) \tag{11}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are trained weights and $\vec{b}_1$ and $\vec{b}_2$ are biases, σ is the sigmoid activation function, N is the number of atoms in the molecule, and $\odot$represents element-wise multiplication.

Attention sum pooling is a simpler variation of the above. By using a softmax function, it performs a weighted average of the atomic environments' contributions

$$\mathbf{X} = \mathrm{softmax}(\mathbf{V}^{(L)}\vec{w}) \tag{12}$$

$$\vec{v}^{(p)} = \sum_i^N \mathbf{X}_i \cdot \vec{v}_i^{(L)} \tag{13}$$

where $\vec{w}$ are trained weights.

After training the model, $\vec{v}^{(p)}$ effectively acts as a machine-learned representation of the molecule that captures only the information about its topology and composition that is useful for predicting the CMC. Finding an optimized representation is a feature of neural networks that happens implicitly during training, called *representation learning*.[59] The final step is a readout neural network: a multilayer perceptron that acts as a nonlinear approximator to map this latent representation vector to the CMC property prediction. Each layer in this neural network, called a "dense" layer, outputs a new vector, $\vec{v}^{(l)}$

$$\vec{v}^{(l)} = \mathbf{W}^{(l)}\vec{v}^{(l-1)} + \vec{b}^{(l)} \tag{14}$$

The full network's architecture is illustrated in Figure 2. The model was implemented using the open-source library Spektral[60] and optimized using an Adam optimizer[61] to minimize the mean-squared error.

A neural network's topology describes the types of layers used, i.e., their functional form and the connection between them. Layers that are parametrized by a weight matrix, **W**, may have different "sizes", meaning that the dimensionality of their output is arbitrary and can be adjusted by changing the dimensions of **W**. The graph layers, dense layers, and the gated attention pool all have this property. These sizes, the type of pooling layer, and the number of each graph and dense layer are all hyperparameters that can be adjusted prior to training. To determine the best combination of hyperparameters for predicting CMCs, an automated searching procedure was employed.

2.2.3. Optimizing GNN Hyperparameters. The Hyperband approach,[62] implemented in Keras Tuner,[63] was used to select a good combination of hyperparameters for the model. Hyperband provides a way to efficiently evaluate the performance of a large search space of hyperparameter configurations. The algorithm assesses several combinations of hyperparameters, initially allocating only a small number of resources to each trial. The hyperparameters for the trials with the best performance are then allocated more resources, while the remainder is discarded. A reduction factor of 3 was chosen, meaning that 2/3 of the trials were discarded after each iteration. This procedure is repeated until the best configuration is found.

The algorithm can be executed multiple times if resources are available to obtain a more reliable result; the training procedure is stochastic, and therefore, the performance of two trials with the same hyperparameters may be different. In this case, a single run was performed. The training data were partitioned into an optimization subset and a validation subset in a ratio of 9:1. The trials were fit to the optimization subset and evaluated based on the RMSE of their predictions on the validation subset. The best hyperparameters determined for the benchmark tasks were then used during the sensitivity analysis.

2.2.4. Adding Uncertainty with a Gaussian Process. To improve the model's reliability, a *surrogate* model was employed that could yield uncertainty estimates alongside CMC predictions. The approach is based on the Convolution-Fed Gaussian Process of Tran et al.[64] The model first computes the latent representation vector, $\vec{v}^{(p)}$, of an input molecule by using a trained GNN. $\vec{v}^{(p)}$ is then standardized, similar to eq 7, but in this case, the standardization applies across each latent feature, n

$$v_n^{(p)'} = \frac{v_n^{(p)} - u_n}{s_n} \tag{15}$$

Again, u_n and s_n were determined from the latent representations of the training molecules' latent representations.

The standardized latent representation vectors of the training data serve as index points for a Gaussian process (GP); see Figure 2. The GP's predicted mean and standard deviation define a predicted normal distribution of a molecule's CMC, $\log X_{\mathrm{cmc}} \sim \mathcal{N}(\mu, \sigma)$.

In this work, the GPs were defined using a Matérn kernel with parameter 1/2 and a fixed noise variance of 1×10^{-5}. Furthermore, the multilayer perceptron component of the GNN used to calculate $\vec{v}^{(p)}$ was employed as the GP's mean function. The kernel parameters were optimized with an Adam optimizer.[61] The same optimization/validation splitting was used as for the GNN hyperparameter search, and training was stopped after 1000 iterations without improvement in the validation predictions' RMSE, or after a total of 5000 iterations. The software implementation was based on GPFlow.[65]

2.2.5. Visualizing the Latent Space. In order to better understand the model's interpretation of chemical space, we exploit the GP's kernel, optimized during training, to plot the molecules in 2D space in a way that respects the model's perception of their "similarity". This approach is inspired by Isayev et al.,[66] who compared the fingerprints of several inorganic compounds to develop so-called materials cartograms. By employing the machine-learned latent representations of molecules, our cartogram reflects molecular similarities learned by the model itself.

Having trained the GP, the learned kernel function was computed between every pair of molecules in the combined data, Complementary and Qin. These kernel values were then normalized within the range [0, 1]. Each molecule was assigned a node in a graph, and these were connected by edges. Each edge was given a weight equal to the normalized kernel value between the two nodes to which it connected.

This was the starting condition for computing a *force-directed graph layout*. The nodes are initialized with a set of 2D coordinates that are uniformly distributed, and they are then moved according to forces acting upon them, primarily an attractive force that acts along the edges. In addition, there are pairwise repulsive forces acting among the nodes that serve to ensure that the equilibrium distance between two nodes is nonzero. In this work, the Force Atlas 2 algorithm was employed.[67,68]

3. RESULTS

3.1. ECFP Feature Selection. The number of atomic environments remaining after each stage of the feature selection process is reported in Table 3. Notably, the ratio of the number of features to the size of the training data set is similar for both tasks (∼74%) and so is the ratio of the initial number of features to the number of selected features (31−33%). The number of features is large compared to many of the empirical models described above but not compared to the number of parameters for the GNN, which is significantly larger. Because this model also aims to cover a large part of the chemical space, a large number of parameters are to be expected.

Table 3. Number of Atomic Environments at Each Stage of the ECFP Feature Selection Process

	number of training data atomic environments		
task	initially	found in multiple molecules	with non-negligible weight
Qin-nonionics	260	201	81
Qin-all	410	302	134

The OPTICS clustering routine on the Qin-All data set using these fingerprints resulted in 24 classes and 31 outlying molecules. The resulting classifications for each molecule are available in the Supporting Information.

3.2. Hyperband Tuning. 725 trials were conducted for each of the Qin training data sets. The best hyperparameters discovered for each set are described in Table 4.

3.3. Benchmark Model Performance. The performances of all of the trained models on the benchmark tasks are reported in Table 5. All of the models developed here outperformed those of Qin et al.[38] For every task, the most accurate model was either the GNN or combined GNN with GP (GNN/GP). The linear model's performance is surprisingly good, considering its relative simplicity, faster optimization, and the far smaller number of parameters it constitutes.

As expected, the performance of all models on the Complementary data set is significantly worse than on the test data. This supports the hypothesis that the Complementary data set molecules are outside of the applicability domain of the models developed here, but it does not exclude the possibility that the models are instead overfitted. Therefore, a more detailed analysis of the applicability domain is provided below.

Table 4. Best Hyperparameters Discovered during Searching[a].

	best value for	
hyperparameter	Qin-nonionics	Qin-all
# graph layers	2	3
H_{G1}	320	64
H_{G2}	256	64
H_{G3}		128
pooling layer	mean pool	sum pool
H_P		
# dense layers	2	2
H_{D1}	128	256
H_{D2}		

[a] The H values refer to the dimensions of the corresponding layer, see Figure 2. Values for H_{G3} and H_{D2} have been omitted where the layers were not included in the model, and the values of H_P were only independent for the gated attention pool so that they are omitted here as well.

Table 5. Benchmark Task Evaluation Results for the Models Trained in This Work Versus Those of the Previous Work[a]

	test RMSE (log μM)		
model	Qin-nonionics	Qin-all	complementary
previous work[38]	0.23	0.30	
ECFP	0.19	0.26	1.57
GNN	$\underline{0.15}$	0.29	1.35
GNN/GP	1.38	$\underline{0.21}$	$\underline{1.32}$

[a] The best RMSE for each task is underlined.

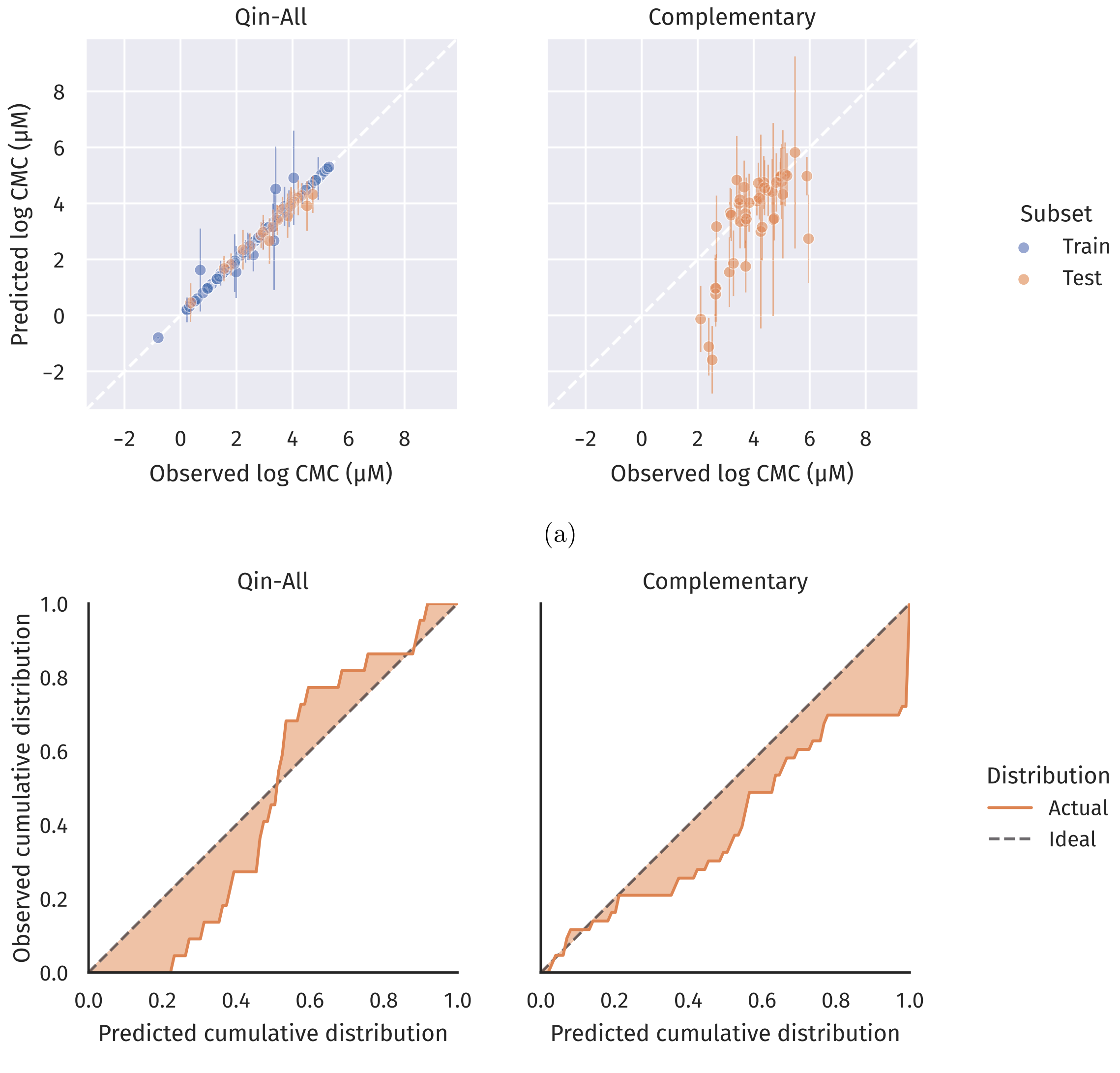
Finally, it is noted that the GNN/GP model's predictive performance on the Qin-Nonionics task was very poor. This indicates that the spacing between the molecules' latent representation vectors, determined from the corresponding GNN, was not a good indicator of similarity, with respect to CMC prediction.

3.3.1. Uncertainty Quantification. The RMSE alone does not capture the quality of the predicted standard deviations. One metric that captures these is the negative log-likelihood (NLL) of observing the true CMCs, given the model's predicted normal distributions

$$\mathrm{NLL} = -\sum_{n} \log p_n(\hat{y}_n) \tag{16}$$

where subscript n is the index of the data, $\hat{y}_n$ is the true CMC value, and p_n is the probability density function of the normal distribution $\mathcal{N}(\mu_n, \sigma_n)$, where μ_n and σ_n are the predicted mean and standard deviation. This metric indicates the relative performance of different models on the same data. (Note that its value scales with the size of the data.) It does not give a good indication of the quality of any individual model in isolation, however. The NLL values are included in the Supporting Information for comparison against future work.

To assess the models' quality individually, the predictions can be visualized against the true CMCs in a parity plot; see Figure 3a. Alternatively, a calibration plot can be used, which compares the cumulative distribution of the residuals against the expected distribution, given a model's predicted standard deviations. The expected distribution stipulates what would be observed if the residuals were drawn from the distributions



(a)

(b)

Figure 3. (a) Parity plots of the predicted CMCs from the GNN/GP model and 95% confidence intervals for the Qin-All and Complementary data sets and (b) corresponding calibration plots for the test data predictions. The "ideal" distribution line indicates the cumulative distribution that would be obtained if residuals were drawn from the model's predicted distributions.

predicted by the corresponding model. Deviations from this distribution indicate whether the model was over- or underconfident (c.f. Tran et al.[64]). The calibration plots are shown in Figure 3b.

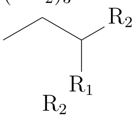
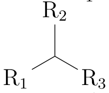
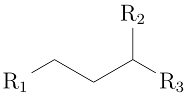
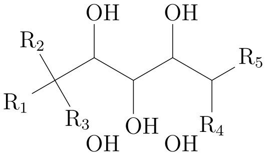
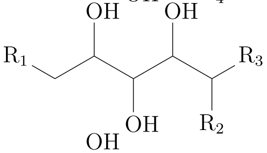
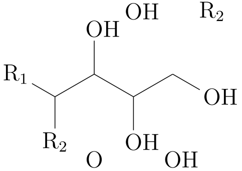
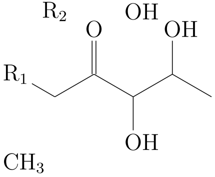
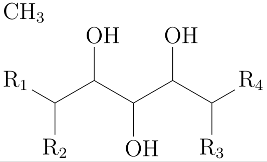
The S-shaped calibration curve for the Qin-All test data indicates that the model was underconfident in its predictions. In fact, there is a spike in the number of observed residuals that are close to the center of the distribution. The corresponding parity plot shows that the predicted uncertainties were relatively small. The Complementary data set calibration curve's asymmetry indicates its tendency to underestimate the CMCs. It shows a generally good agreement with the ideal distribution, with the greatest discrepancy above 0.8, which indicates that there are several surfactants whose CMC predictions are too low and that these predictions are overconfident, i.e., their predicted standard deviations are too small.

3.4. ECFP interpretations. The weights of the ECFP models are coefficients corresponding to the scaled counts of the selected atomic environments. Referring to eqs 5 and 7, these coefficients indicate the change in a predicted CMC when the count of $\mathcal{E}_m$ increases by s_m from its average, u_m. A more readily interpreted value can be achieved by rescaling the coefficient, w_m, for an environment:

$$w'_m = \frac{w_m(1 - u_m)}{s_m} \tag{17}$$

which indicates the difference in predicted CMC between a molecule containing one $\mathcal{E}_m$ and a molecule without any $\mathcal{E}_m$, but which otherwise contains exactly the same number as all the other environments. This scaled weight can be interpreted as a rough indication of the relative importance of different environments to determining CMC; "rough" because it may not be physically plausible that two molecules exist that are distinguished only by the number of $\mathcal{E}_m$ that they contain. This is particularly true for larger environments that envelope smaller ones. The largest scaled weights for the two ECFP models are given in Table 6.

Table 6. Atomic Environments with the Greatest Importance to the CMC According to the Trained ECFP Models

Qin-All		Qin-Nonionics	
Environment	Scaled weight	Environment	Scaled Weight
$(CH_2)_5$	−0.64	$(CH_2)_5$	−0.76
$(CH_2)_3$	−0.55	$(CH_2)_3$	−0.69
Cl^-	0.31		−0.29
Br^-	0.29		−0.25
	−0.27		−0.19
CH_2	−0.23		0.14
$R_1{-}O{-}R_2$	0.18		−0.12
OH	−0.17		0.09
$R_1{-}O{-}(CH_2)_2OH$	−0.14	CH_3	−0.06
$CH_2{-}O{-}(CH_2)_2OH$	−0.14		0.05

Both the Qin-Nonionic and Qin-All models agree that alkyl chain environments constitute the top two most important contributors to the CMC, suggesting that tail length is the most important factor. The model trained on all surfactant classes includes two counterions in its most important environments: Cl^- and Br^-. This is to be expected; ionic surfactants typically have much larger CMCs than nonionics, and the model appears to distinguish these by their counterion. The Qin-Nonionics model identifies environments from the headgroups of sugar-based surfactants as being important. These surfactant headgroups possessed relatively complex topologies and, therefore, several environments; it may have been necessary for the model to use many of these environments in order to accurately distinguish between their CMCs.

3.5. Applicability Domain Analysis. Several of the molecules included in the Complementary data set were expected to be outside of the applicability domain, which justifies their poor prediction accuracy. The majority of the outliers' CMCs are underpredicted, and the GNN/GP model is overconfident in their predictions. The 13 surfactants whose predictions' residuals are greater than the 95% confidence interval (CI) are shown in Figure 4. These constitute 3 nonionic surfactants, 1 zwitterionic surfactant, and 9 ionic surfactants. Of the ionic surfactants, 5 have quaternary ammonium salts (quats) as counterions.

It is useful to examine the types of counterions in these molecules compared to the training data. In the training data, there are no examples with a bromate, nitrate, or carboxylate counterion, but there are two examples of a quat counterion. However, the quats in the training data are isotropic: tetrapropylammonium and tetramethylammonium. In contrast, those that are underpredicted are highly anisotropic and are effectively surfactants themselves; the behavior of these compounds, in terms of CMC, can be expected to be very different from that of the surfactants in the training data. Furthermore, the two polyoxyethylene alkyl ethers in Figure 4 have remarkably small tail groups relative to the examples in the training data, which justifies their erroneous CMC predictions. This leaves two surfactants among the outliers that might reasonably be expected to lie within the applicability domain, constituting 4.7% of the total Complementary data set, which is close to the 5% that are expected to be outside the 95% CI.

To gain insight into the relationships between molecules, we exploited the kernel function from the trained GP to create a molecular cartogram, as explained in Section 2. The resulting cartogram is shown in Figure 5. The Complementary

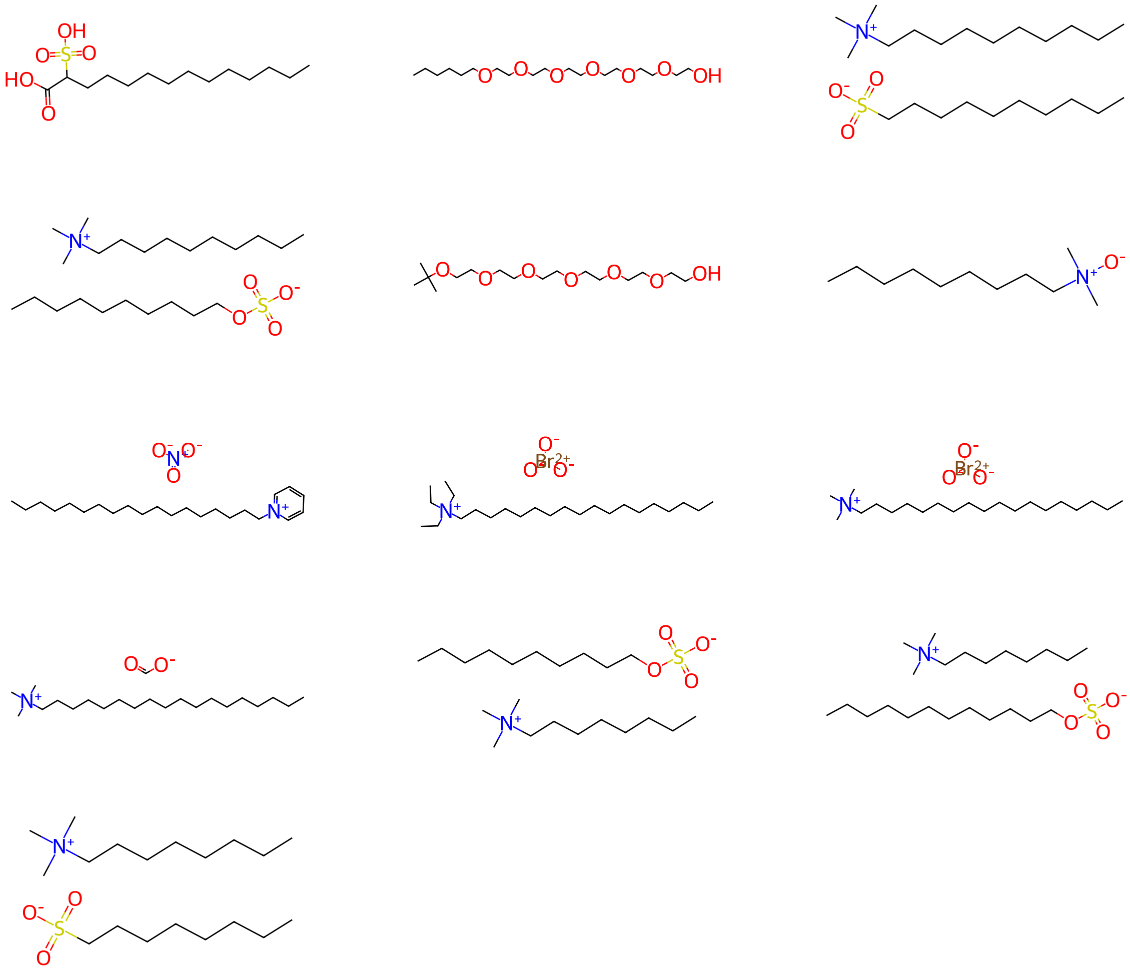


Figure 4. 13 surfactants and counterions in the Complementary data set with residuals that were greater than the 95% confidence interval.

molecules whose residuals are above the 95% CI are also highlighted separately.

From the cartogram of the entire data, it is apparent that the majority of the surfactants are segregated on the basis of the type of counterion in the solution. This suggests that the model has learned that the counterion is an important factor for determining the CMC and that the weights associated with this counterion have a profound impact on the CMC prediction. However, the visualization of the Complementary molecules with CMCs above 95% CI indicates that the model classifies all of these molecules as being primarily similar to the nonionic or zwitterionic compounds. This is clearly erroneous for the ionic compounds: the quats, the bromates, the carboxylates, and the nitrates. This suggests that the model has not learned appropriate weights for these counterions; hence, it fails to properly segregate them.

3.6. Sensitivity Analysis. Figure 6 shows the test set evaluations for the GNN and ECFP models using repeated stratified *k*-fold cross-validation. The ECFP and GNN models show very similar performance on the Qin-All data set, but the ECFP model outperforms the GNN model for all training ratios on the Qin-Nonionics data set. This is indicative of the propensity of neural networks to overfit, especially on such small data set sizes. Notably, there is also a much broader distribution of RMSEs on the Qin-Nonionics data. This suggests that both of the Qin-Nonionics models are much more sensitive to the specific molecules that are included in the train and test splits.

Extrapolating the trend lines indicates that the benchmark performance is much better than what would be expected. This highlights the importance of selecting an appropriate training data split that spans the entirety of the chemical space of interest. On a data set as small as the one used here, it is not possible to achieve this without using a high training ratio.

4. DISCUSSION

The ideal molecular representation of an active chemical depends on the task at hand. Ideally, it should be compact but complete;[69,70] 'as simple as possible, but not simpler.' To that end, the representation should contain enough information to distinguish between isomers with distinct properties. However, concessions can be made if we restrict the model's domain and self-imposed limits on the type of isomers we expose the model to, both during training and in use. Representations may also include descriptions of state, such as temperature and pressure,[71] but this is redundant in cases where the training data span a very limited range of states.

The representations employed by both the GNN and the linear models capture topological information, and the performances of all of the models on in-domain data suggest that this is sufficient for the task of predicting CMC very accurately. However, both models are unable to distinguish between certain positional isomers depending on the size of

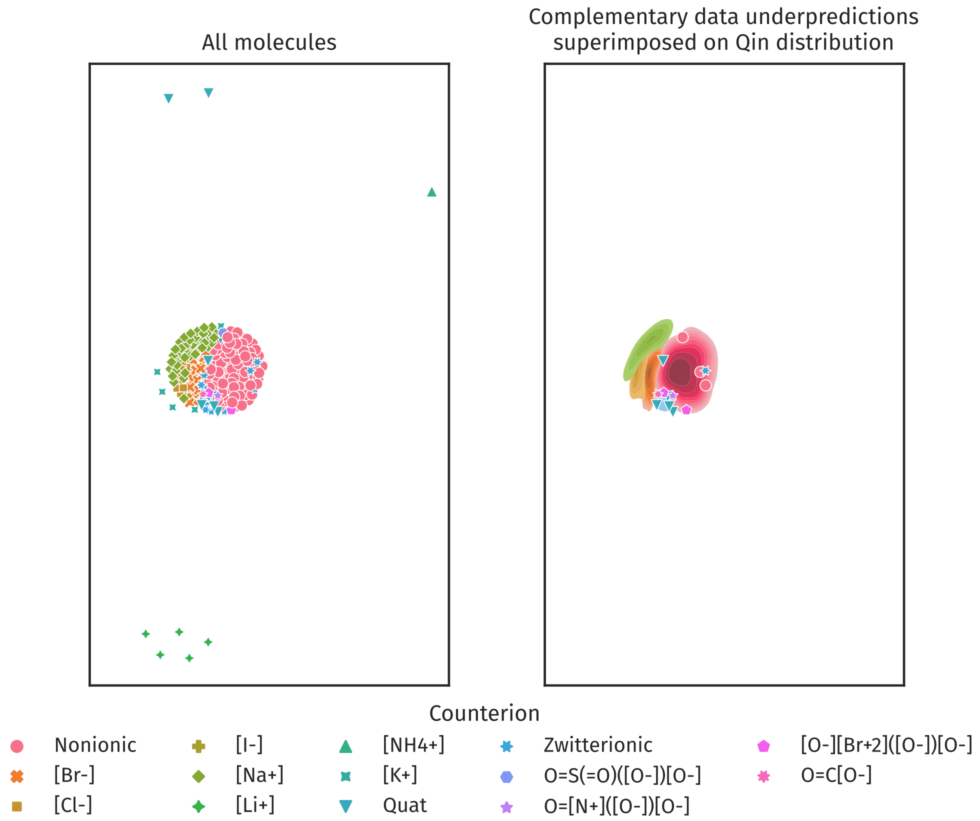


Figure 5. Cartogram of the molecules in the combined Complementary and Qin data sets using a force-directed graph layout. Left: The entirety of the combined Complementary and Qin data sets, where each molecule is assigned a point. Right: The Complementary molecules that are overconfidently underpredicted, so that the residual is above the 95% confidence interval, are superimposed on a kernel density estimate (KDE) plot of the Qin data. This KDE plot is an estimate of the distribution of the Qin molecules on the left, colored by the type of counterion.

the atomic environments that they consider. In the case of ECFPs, this is dictated by the radius around each atom that is included in the fingerprint, while for the GNNs, this is determined by the number of consecutive graph layers.

Increasing these parameters both increases computational cost and model complexity, introducing more weights and therefore requires more data to optimize them appropriately. The sensitivity analysis demonstrated that this increase in complexity also increases the propensity for overfitting; however, the benchmarking results demonstrate that using a proper selection of training samples can yield more accurate models. In cases where there are fewer samples available and some chemical classes are poorly represented in the training data, the simpler, linear model may be preferable.

One of the great advantages of using such a topological approach is that the contributions of each molecular fragment can be explicitly determined, as shown in the section on ECFP interpretations. In the case of the GNN, introducing a kernel to the model, via a Gaussian process operating on the GNN's learned latent space representations, offers a quantitative measure of molecular similarity that can simultaneously be employed for adding uncertainty to the CMC predictions and visualizing the chemical space of the training data. Superimposing the test data onto this cartogram highlights which molecules may be associated with erroneous predictions, based on the fact that they are clustered among training data molecules with very different chemistries.

Case-by-case examination of these molecules highlights the nature of these chemical differences, which can be related to properties important for micellization. Broadly speaking, the model failed in three cases:

- Where ionic effects of the solute were not learned, as in the case of the counterions that were unique to the Complementary data set.
- When there was a significant difference in the strength of the hydrophobic interactions, as in the case of the surfactants with very small tail groups.
- When the counterion was itself surfactant-like, as in the case of the ammonium quaternary ions, which implies the system should better be described as a binary mixture of surfactants. Notably, the models developed here were not trained against predictions of CMC for mixed surfactant systems.

These results stress the importance of applying domain knowledge in developing and analyzing the results of deep learning models. The uncertainty quantification is unreliable when the systems' behavior is starkly different from what the model can expect to learn from the training data.

Future efforts may consider incorporating another term in the loss function for the GNN that explicitly biases the model toward learning a form of $\vec{v}^{(p)}$ that captures chemical similarity based on user-defined metrics; for example, the length of the carbon tail. This approach would enable chemical knowledge to be explicitly encoded within the model and may correct for some of the aforementioned failure cases. Alternatively, a variational Gaussian process could be used, which approximates the Gaussian process using a fixed-size set of "pseudopoints";[72] this would enable the entire GNN/GP

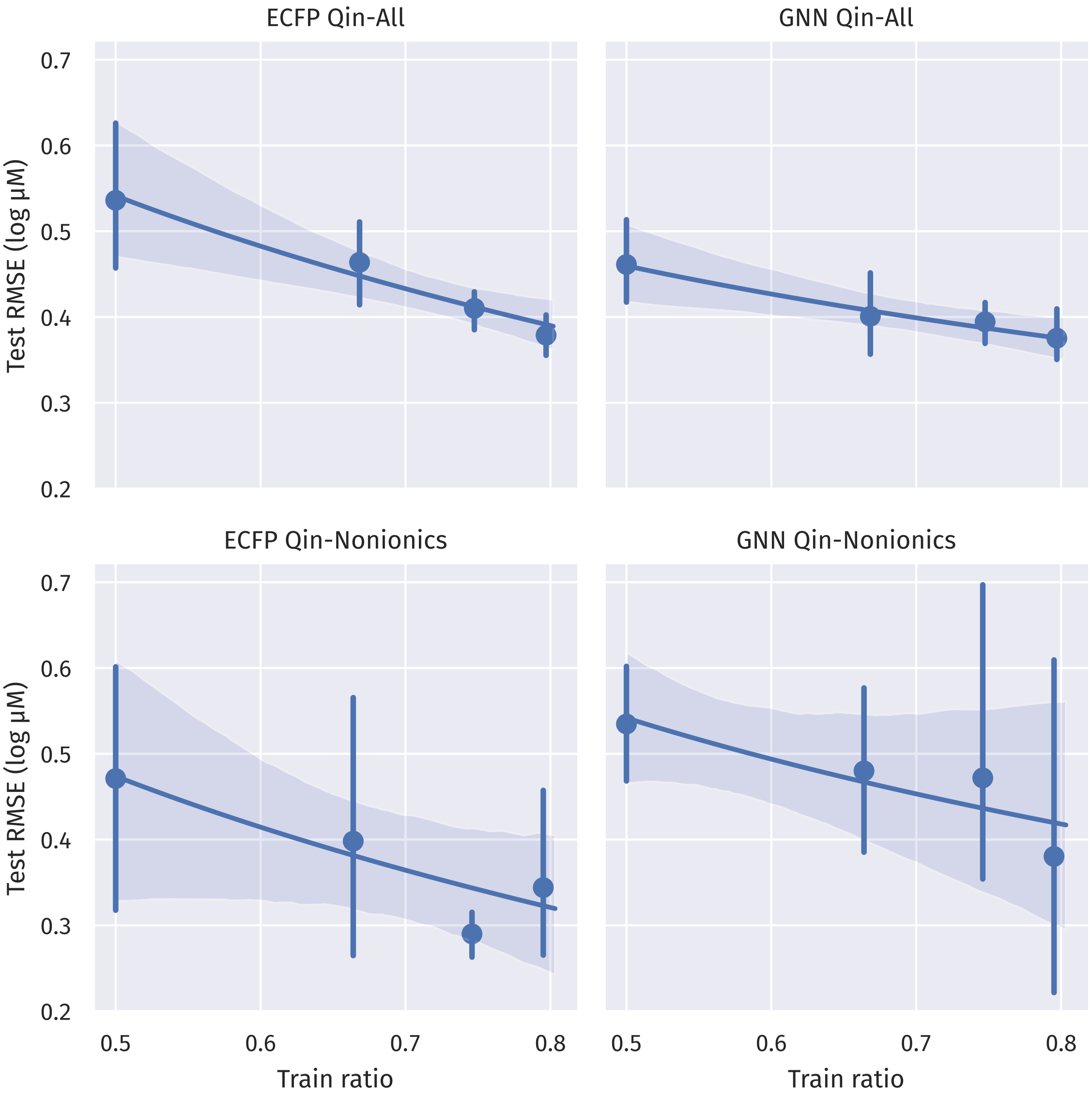


Figure 6. Test set RMSEs for the sensitivity analysis models. The average RMSE for each value of k is indicated as well as the range of the RMSEs. A logarithmic fit to the data is shown, with a 95% confidence interval determined by bootstrapping using 1000 repeats.

model to be trained at once using backpropagation and can be applied when the training data size is larger.[73]

5. CONCLUSIONS

Empirical models were developed and applied to predict CMCs from two data sets of aqueous surfactants. One data set was partitioned into training and test data (Qin-All), and a subset of the nonionic surfactants within this data was also used as a separate prediction task (Qin-Nonionics). The Complementary data set was collected from a different source and contained molecules with somewhat different chemistries than the above.

A linear model based on ECFPs demonstrated remarkably good performance, improving on a previous work[38] that applied a more complex GNN model despite using a smaller number of parameters and having a much faster optimization time. A new model was presented that improved the architecture of previous work's GNN using a hyperparameter search algorithm, which was capable of obtaining better performances than the ECFP model on the Qin-Nonionics task and demonstrated a better ability to generalize to the Complementary data set.

Sensitivity analysis showed that our GNN models have a tendency to overfit when training data samples do not adequately cover the chemical space of interest. When using small data sets, with only a few examples of certain surfactant classes, our analysis suggests that it may be preferable to use a simpler functional form, like the linear ECFP model.

A surrogate model was developed by feeding the latent space representation of a molecule, learned by the GNN model, to a Gaussian process. This yielded uncertainty estimates, alongside CMC predictions. Although this model failed when applied to the Qin-Nonionics task, it yielded the best predictive performance of all of the models trained here for the Qin-All task as well as providing good uncertainty estimates on the in-domain Complementary test data. This approach would allow practitioners to gauge their confidence in the model's predictions for systems within the applicability domain.

Finally, the kernel function learned while training the Gaussian process was employed to visualize the chemical space by using a molecular cartogram. By analysis of this cartogram, it was shown that chemical intuition could be employed to determine which molecules were likely poorly represented in the latent space, based on the fact that they were surrounded by molecules with different chemistries. This promises to be a useful technique for exploring the limits of a model's applicability domain, as well as understanding why the model yields its predictions for a given molecule based on proximity to training set molecules within the cartogram.

This work demonstrates the potential of Gaussian processes to add uncertainty quantification to machine-learning models with minimal overhead. There is still scope to overcome the limitations of these models with respect to small data sets, such as Qin-Nonionics, and out-of-domain molecules, which could be achieved by explicitly biasing the latent space.

ASSOCIATED CONTENT

Data Availability Statement

Source code for featurization and model training, graph neural network logs, metrics for hyperparameter optimization and final training, and individual model predictions are available at https://github.com/a-ws-m/CaMCaNN/.

AUTHOR INFORMATION

Corresponding Author

Alberto Striolo − *Department of Chemical Engineering, University College London, London WC1E 7JE, U.K.; School of Sustainable Chemical, Biological and Materials Engineering, University of Oklahoma, Norman, Oklahoma 73019-0390, United States;* orcid.org/0000-0001-6542-8065; Email: astriolo@ou.edu

Authors

Alexander Moriarty − *Department of Chemical Engineering, University College London, London WC1E 7JE, U.K.;* orcid.org/0000-0001-7525-1419

Takeshi Kobayashi − *Department of Chemical Engineering, University College London, London WC1E 7JE, U.K.;* orcid.org/0000-0002-4823-1567

Matteo Salvalaglio − *Department of Chemical Engineering, University College London, London WC1E 7JE, U.K.;* orcid.org/0000-0003-3371-2090

Panagiota Angeli − *Department of Chemical Engineering, University College London, London WC1E 7JE, U.K.*

Ian McRobbie − *Innospec Ltd., Ellesmere Port CH65 4EY, U.K.*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jctc.3c00868

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

A.M. was supported by the EPSRC and SFI Center for Doctoral Training in the Advanced Characterization of Materials (grant number EP/S023259/1) and the ABK studentship. This work was supported in part by a UK Research and Innovation grant (EPSRC project number EP/V032909/1). A.S. gratefully acknowledges financial support from the Asahi Glass Chair of Chemical Engineering at the University of Oklahoma. The authors acknowledge the use of the UCL Myriad High Performance Computing Facility (Myriad@UCL), and associated support services, in the completion of this work. All the authors are grateful to Innospec Ltd. for their support.

REFERENCES

(1) Rosen, M. J.; Kunjappu, J. T. *Surfactants and Interfacial Phenomena*; John Wiley & Sons, 2012.

(2) Klevens, H. B. Structure and Aggregation in Dilate Solution of Surface Active Agents. *J. Am. Oil Chem. Soc.* **1953**, *30*, 74−80.

(3) Puvvada, S.; Blankschtein, D. Molecular-thermodynamic Approach to Predict Micellization, Phase Behavior and Phase Separation of Micellar Solutions. I. Application to Nonionic Surfactants. *J. Chem. Phys.* **1990**, *92*, 3710−3724.

(4) de Miguel, R.; Rubí, J. M. Gibbs Thermodynamics and Surface Properties at the Nanoscale. *J. Chem. Phys.* **2021**, *155*, No. 221101.

(5) Frenkel, D.; Smit, B. *Molecular Simulation: From Algorithms to Applications*; Elsevier, 2001.

(6) Joshi, S. Y.; Deshmukh, S. A. A Review of Advancements in Coarse-Grained Molecular Dynamics Simulations. *Mol. Simul.* **2021**, *47*, 786−803.

(7) Filipe, H. A. L.; Loura, L. M. S. Molecular Dynamics Simulations: Advances and Applications. *Molecules* **2022**, *27*, 2105.

(8) Jorge, M. Molecular Dynamics Simulation of Self-Assembly of n-Decyltrimethylammonium Bromide Micelles. *Langmuir* **2008**, *24*, 5714−5725.

(9) Jusufi, A.; Panagiotopoulos, A. Z. Explicit- and Implicit-Solvent Simulations of Micellization in Surfactant Solutions. *Langmuir* **2015**, *31*, 3283−3292.

(10) Fitzgerald, G.; DeJoannis, J.; Meunier, M. *Modeling, Characterization, and Production of Nanomaterials*; Tewary, V. K.; Zhang, Y., Eds.; Woodhead Publishing Series in Electronic and Optical Materials; Woodhead Publishing, 2015; pp 3−53.

(11) Khedr, A.; Striolo, A. Quantification of Ostwald Ripening in Emulsions via Coarse-Grained Simulations. *J. Chem. Theory Comput.* **2019**, *15*, 5058−5068.

(12) Bochicchio, D.; Salvalaglio, M.; Pavan, G. M. Into the Dynamics of a Supramolecular Polymer at Submolecular Resolution. *Nat. Commun.* **2017**, *8*, No. 147.

(13) Anderson, R. L.; Bray, D. J.; Del Regno, A.; Seaton, M. A.; Ferrante, A. S.; Warren, P. B. Micelle Formation in Alkyl Sulfate Surfactants Using Dissipative Particle Dynamics. *J. Chem. Theory Comput.* **2018**, *14*, 2633−2643.

(14) Khedr, A.; Striolo, A. DPD Parameters Estimation for Simultaneously Simulating Water−Oil Interfaces and Aqueous Nonionic Surfactants. *J. Chem. Theory Comput.* **2018**, *14*, 6460−6471.

(15) Santos, A. P.; Panagiotopoulos, A. Z. Determination of the Critical Micelle Concentration in Simulations of Surfactant Systems. *J. Chem. Phys.* **2016**, *144*, No. 044709.

(16) Nivón-Ramírez, D.; Reyes-García, L. I.; Oviedo-Roa, R.; Gómez-Balderas, R.; Zuriaga-Monroy, C.; Martínez-Magadán, J.-M. Critical Micelle Concentration of SDS through DPD Simulations Using COSMO-RS−Based Interaction Parameters, the Thermal Effects. *Colloids Surf., A* **2022**, *645*, No. 128867.

(17) Klamt, A.; Schüürmann, G. COSMO: A New Approach to Dielectric Screening in Solvents with Explicit Expressions for the Screening Energy and Its Gradient. *J. Chem. Soc., Perkin Trans. 2* **1993**, *0*, 799−805.

(18) Klamt, A.; Eckert, F.; Arlt, W. COSMO-RS: An Alternative to Simulation for Calculating Thermodynamic Properties of Liquid Mixtures. *Annu. Rev. Chem. Biomol. Eng.* **2010**, *1*, 101−122.

(19) Turchi, M.; Karcz, A. P.; Andersson, M. P. First-Principles Prediction of Critical Micellar Concentrations for Ionic and Nonionic Surfactants. *J. Colloid Interface Sci.* **2022**, *606*, 618−627.

(20) Klamt, A.; Huniar, U.; Spycher, S.; Keldenich, J. COSMOmic: A Mechanistic Approach to the Calculation of Membrane-Water Partition Coefficients and Internal Distributions within Membranes and Micelles. *J. Phys. Chem. B* **2008**, *112*, 12148−12157.

(21) Jakobtorweihen, S.; Yordanova, D.; Smirnova, I. Predicting Critical Micelle Concentrations with Molecular Dynamics Simulations and COSMOmic. *Chem. Ing. Tech.* **2017**, *89*, 1288−1296.

(22) Klamt, A.; Schwöbel, J.; Huniar, U.; Koch, L.; Terzi, S.; Gaudin, T. COSMOplex: Self-Consistent Simulation of Self-Organizing Inhomogeneous Systems Based on COSMO-RS. *Phys. Chem. Chem. Phys.* **2019**, *21*, 9225−9238.

(23) Herbert, J. M. Dielectric Continuum Methods for Quantum Chemistry. *WIREs Comput. Mol. Sci.* **2021**, *11*, No. e1519.

(24) Eckert, F.; Klamt, A. Fast Solvent Screening via Quantum Chemistry: COSMO-RS Approach. *AIChE J.* **2002**, *48*, 369−385.

(25) Li, X.-S.; Lu, J.-F.; Li, Y.-G.; Liu, J.-C. Studies on UNIQUAC and SAFT Equations for Nonionic Surfactant Solutions. *Fluid Phase Equilib.* **1998**, *153*, 215−229.

(26) Cheng, J.-S.; Chen, Y.-P. Correlation of the Critical Micelle Concentration for Aqueous Solutions of Nonionic Surfactants. *Fluid Phase Equilib.* **2005**, *232*, 37−43.

(27) Voutsas, E. C.; Flores, M. V.; Spiliotis, N.; Bell, G.; Halling, P. J.; Tassios, D. P. Prediction of Critical Micelle Concentrations of Nonionic Surfactants in Aqueous and Nonaqueous Solvents with UNIFAC. *Ind. Eng. Chem. Res.* **2001**, *40*, 2362−2366.

(28) Veerasamy, R.; Rajak, H.; Jain, A.; Sivadasan, S.; Varghese, C. P.; Agrawal, R. K. Validation of QSAR Models-Strategies and Importance. *Int. J. Drug Des. Discovery* **2011**, *3*, 511−519.

(29) Tropsha, A. Best Practices for QSAR Model Development, Validation, and Exploitation. *Mol. Inf.* **2010**, *29*, 476−488.

(30) Leonard, J. T.; Roy, K. On Selection of Training and Test Sets for the Development of Predictive QSAR Models. *QSAR Comb. Sci.* **2006**, *25*, 235−251.

(31) Gantzer, P.; Creton, B.; Nieto-Draghi, C. Inverse-QSPR for de Novo Design: A Review. *Mol. Inf.* **2020**, *39*, No. 1900087.

(32) Bolboaca, S. D.; Jantschi, L.; Diudea, M. V. Molecular Design and QSARs/QSPRs with Molecular Descriptors Family. *Curr. Comput.-Aided Drug Des.* **2013**, *9*, 195−205.

(33) Zefirov, N. S.; Palyulin, V. A. Fragmental Approach in QSPR. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 1112−1122.

(34) Mattei, M.; Kontogeorgis, G. M.; Gani, R. Modeling of the Critical Micelle Concentration (CMC) of Nonionic Surfactants with an Extended Group-Contribution Method. *Ind. Eng. Chem. Res.* **2013**, *52*, 12236−12246.

(35) Gani, R.; Harper, P. M.; Hostrup, M. Automatic Creation of Missing Groups through Connectivity Index for Pure-Component Property Prediction. *Ind. Eng. Chem. Res.* **2005**, *44*, 7262−7269.

(36) Li, J.; Cheng, K.; Wang, S.; Morstatter, F.; Trevino, R. P.; Tang, J.; Liu, H. Feature Selection: A Data Perspective. *ACM Comput. Surv.* **2017**, *50*, 1−45.

(37) Guyon, I.; Elisseeff, A. An Introduction to Variable and Feature Selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157−1182.

(38) Qin, S.; Jin, T.; Van Lehn, R. C.; Zavala, V. M. Predicting Critical Micelle Concentrations for Surfactants Using Graph Convolutional Neural Networks. *J. Phys. Chem. B* **2021**, *125*, 10610−10620.

(39) Bejani, M. M.; Ghatee, M. A Systematic Review on Overfitting Control in Shallow and Deep Neural Networks. *Artif. Intell. Rev.* **2021**, *54*, 6391−6438.

(40) Louis, S.-Y.; Zhao, Y.; Nasiri, A.; Wang, X.; Song, Y.; Liu, F.; Hu, J. Graph Convolutional Neural Networks with Global Attention for Improved Materials Property Prediction. *Phys. Chem. Chem. Phys.* **2020**, *22*, 18141−18148.

(41) Rittig, J. G.; Gao, Q.; Dahmen, M.; Mitsos, A.; Schweidtmann, A. M. Graph Neural Networks for the Prediction of Molecular Structure-Property Relationships, arXiv:2208.04852v1. arXiv.org e-Print archive. http://arxiv.org/abs/2208.04852 (accessed July 25, 2022).

(42) Singh, K.; Münchmeyer, J.; Weber, L.; Leser, U.; Bande, A. Graph Neural Networks for Learning Molecular Excitation Spectra. *J. Chem. Theory Comput.* **2022**, *18*, 4408−4417.

(43) Dietrich, F.; Advincula, X. R.; Gobbo, G.; Bellucci, M.; Salvalaglio, M. Machine Learning Nucleation Collective Variables with Graph Neural Networks. ChemRxiv, https://chemrxiv.org/engage/chemrxiv/article-details/649e32956e1c4c986b909355, 2023.

(44) Errica, F.; Giulini, M.; Bacciu, D.; Menichetti, R.; Micheli, A.; Potestio, R. A Deep Graph Network−Enhanced Sampling Approach to Efficiently Explore the Space of Reduced Representations of Proteins. *Front. Mol. Biosci.* **2021**, *8*, No. 637396, DOI: 10.3389/fmolb.2021.637396.

(45) Wellawatte, G. P.; Gandhi, H. A.; Seshadri, A.; White, A. D. A Perspective on Explanations of Molecular Prediction Models. *J. Chem. Theory Comput.* **2023**, *19*, 2149−2160.

(46) Mukerjee, P.; Mysels, K. J.*Critical Micelle Concentrations of Aqueous Surfactant Systems*, 1971; pp 51−65.

(47) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.

(48) Merkwirth, C.; Lengauer, T. Automatic Generation of Complementary Descriptors with Molecular Graph Networks. *J. Chem. Inf. Model.* **2005**, *45*, 1159−1168.

(49) Tanimoto, T. T. *Elementary Mathematical Theory of Classification and Prediction*; IBM Internal Report, 1958.

(50) Bajusz, D.; Rácz, A.; Héberger, K. Why Is Tanimoto Index an Appropriate Choice for Fingerprint-Based Similarity Calculations? *J. Cheminf.* **2015**, *7*, No. 20.

(51) Butina, D. Unsupervised Data Base Clustering Based on Daylight's Fingerprint and Tanimoto Similarity: A Fast and Automated Way To Cluster Small and Large Data Sets. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 747−750.

(52) Ankerst, M.; Breunig, M. M.; Kriegel, H.-P.; Sander, J. OPTICS: Ordering Points to Identify the Clustering Structure. *ACM SIGMOD Record* **1999**, *28*, 49−60.

(53) Zou, H.; Hastie, T. Regularization and Variable Selection via the Elastic Net. *J. R. Stat. Soc. Ser. B: Stat. Methodol.* **2005**, *67*, 301−320.

(54) Efron, B.; Hastie, T.; Johnstone, I.; Tibshirani, R. Least Angle Regression. *Ann. Stat.* **2004**, *32*, 407−499.

(55) McDonald, G. C. Ridge Regression. *WIREs Comput. Stat.* **2009**, *1*, 93−100.

(56) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825−2830.

(57) Kipf, T. N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks, arXiv:1609.02907v4. arXiv.org e-Print archive. http://arxiv.org/abs/1609.02907 (accessed Sept 9, 2017).

(58) Li, Y.; Tarlow, D.; Brockschmidt, M.; Zemel, R. Gated Graph Sequence Neural Networks, arXiv:1511.05493v4. arXiv.org e-Print archive. http://arxiv.org/abs/1511.05493 (accessed Nov 17, 2017).

(59) Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press, 2016; pp 524−554.

(60) Grattarola, D.; Alippi, C. Graph Neural Networks in TensorFlow and Keras with Spektral, arXiv:2006.12138v. arXiv.org e-Print archive. http://arxiv.org/abs/2006.12138 (accessed June 22, 2020).

(61) Kingma, D. P.; Ba, J. Adam: A Method for Stochastic Optimization., arXiv:1412.6980v9. arXiv.org e-Print archive. http://arxiv.org/abs/1412.6980 (accessed Dec 22, 2017).

(62) Li, L.; Jamieson, K.; DeSalvo, G.; Rostamizadeh, A.; Talwalkar, A. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. *J. Mach. Learn. Res.* **2018**, *18*, 1−52.

(63) Chollet, F.et al. Keras: The Python Deep Learning Library. *Astrophysics Source Code Library*, 2018, ascl:1806.022.

(64) Tran, K.; Neiswanger, W.; Yoon, J.; Zhang, Q.; Xing, E.; Ulissi, Z. W. Methods for Comparing Uncertainty Quantifications for Material Property Predictions. *Mach. Learn.: Scie. Technol.* **2020**, *1*, No. 025006.

(65) Matthews, A. G. d. G.; van der Wilk, M.; Nickson, T.; Fujii, K.; Boukouvalas, A.; León-Villagrá, P.; Ghahramani, Z.; Hensman, J. GPflow: AGaussian Process Library Using TensorFlow. *J. Mach. Learn. Res.* **2017**, *18*, 1−6.

(66) Isayev, O.; Fourches, D.; Muratov, E. N.; Oses, C.; Rasch, K.; Tropsha, A.; Curtarolo, S. Materials Cartography: Representing and Mining Materials Space Using Structural and Electronic Fingerprints. *Chem. Mater.* **2015**, *27*, 735−743.

(67) Jacomy, M.; Venturini, T.; Heymann, S.; Bastian, M. ForceAtlas2, a Continuous Graph Layout Algorithm for Handy Network Visualization Designed for the Gephi Software. *PLoS One* **2014**, *9*, No. e98679.

(68) Bastian, M.; Heymann, S.; Jacomy, M. In *Gephi: An Open Source Software for Exploring and Manipulating Networks*, Proceedings of the International AAAI Conference on Web and Social Media; 2009; Vol. 3, pp 361−362.

(69) Faber, F.; Lindmaa, A.; von Lilienfeld, O. A.; Armiento, R. Crystal Structure Representations for Machine Learning Models of Formation Energies. *Int. J. Quantum Chem.* **2015**, *115*, 1094−1101.

(70) Himanen, L.; Jäger, M. O. J.; Morooka, E. V.; Federici Canova, F.; Ranawat, Y. S.; Gao, D. Z.; Rinke, P.; Foster, A. S. DScribe: Library of Descriptors for Machine Learning in Materials Science. *Comput. Phys. Commun.* **2020**, *247*, No. 106949, DOI: 10.1016/j.cpc.2019.106949.

(71) Chen, C.; Ye, W.; Zuo, Y.; Zheng, C.; Ong, S. P. Graph Networks as a Universal Machine Learning Framework for Molecules and Crystals. *Chem. Mater.* **2019**, *31*, 3564−3572.

(72) Hensman, J.; Fusi, N.; Lawrence, N. D. Gaussian Processes for Big Data, arXiv:1309.6835v1. arXiv.org e-Print archive. http://arxiv.org/abs/1309.6835 (accessed Sept 26, 2013).

(73) Moriarty, A.; Morita, K.; Butler, K. T.; Walsh, A. UnlockNN: Uncertainty Quantification for Neural Network Models of Chemical Systems. *J. Open Source Software* **2022**, *7*, 3700.