In too deep? Neural networks versus linear models for critical micelle concentration prediction

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

†*Department of Chemical Engineering, University College London, UK*

‡*Gallolgy College of Engineering, University of Oklahoma, USA*

¶*Senior Vice President, Research and Technology, Innospec Ltd., Ellesmere Port, UK*

E-mail: alexander.moriarty.21@ucl.ac.uk

Abstract

TODO: Write this.

Introduction

The critical micelle concentration (CMC) of a surfactant defines the concentration above which the surfactant monomers self-assemble in solution to form micelles. It is an important property because the formation of micelles affects many interfacial phenomena[1] micelles can encapsulate hydrophobic molecules to form useful complexes, but the formation of micelles also inhibits certain processes, like the aggregation of surfactants at interfaces in order to reduce surface tension.

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

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

where A and B are empirical constants that depend on the temperature and the homologous series, i.e. the headgroup. The model is simple yet accurate, and it 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 only applicable to surfactants with a specific headgroup and a linear carbon tail. One of the goals of the quantitative structure-property relationship (QSPR) development is to produce models that are general so that we can apply them to a diverse range of compounds, design novel molecules with target properties, and interpret the models' results to glean chemical insights.

There have been a wealth of investigations into making more general models for CMC prediction; here, a brief review of some diverse and promising approaches for predicting CMCs of aqueous, single-surfactant systems will be given.

Theoretical approaches have the potential to be the most useful type of predictive model if they are accurate and applicable to the desired system, as they are directly related to scientific knowledge, and their results can be understood in terms of well-studied principles. Puvvada and Blankschtein[3] derived a phenomenological model for studying aqueous nonionic surfactant systems that enabled CMC prediction and modelling other properties across a range of temperatures. The model they developed was the product of decomposing the process of micellisation into discrete steps that they could describe thermodynamically so as to yield a description of the free energy of micellisation in terms of a set of molecular parameters:

- 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 effectively describes the thickness of an 'interaction region' around the tail.[4] This must also be estimated.

A functional form to estimate the parameters was described for linear, nonionic, polyoxyethylene alcohol surfactants. The model attained impressive accuracy for some predictions: a root-mean-squared error (RMSE) of approximately $0.14\log\mu\mathrm{M}$ for the group $C_{10}E_i$, where $i \in [3, 6]$, and $0.21\log\mu\mathrm{M}$ 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 expect that this inaccuracy is because the model overestimates the CMC values for systems in which the micelles do not grow.

The connection this model established between a small set of physically meaningful properties that can be estimated and emergent properties of surfactants 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 only applicable to linear hydrocarbon chains, not branched nor heterogeneous tail groups. Estimating the average cross-sectional area of the head group may also not be trivial.

Greater generalisability and better accuracy can be realised by considering the process of aggregation on a more precise level: identifying the behaviour 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 from these calculations. By combining this information with entropy considerations, statistical mechanics can indicate free energy changes, such as those associated with micellisation, from which the CMC can be derived.

Molecular dynamics (MD) simulations treat individual atoms, in the all-atomistic (AA) approach, or groups of atoms, in the coarse-grained (CG) approach, as particles in a box

that exert force on each other. This allows the particles' movement to be simulated by iteratively updating their positions and momenta in discrete 'time steps'. Jorge[5] used two types of AA approach to simulate the self-assembly of n-decyltrimethylammonium bromide: one approach that explicitly considered hydrogens, and another, united-atom approach that grouped hydrogens with their respective connecting atom. 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. The more accurate model for determining CMC, the united-atom approach, yielded a CMC with an error of $0.23\log\mu M$. The author asserted that the simulation size, i.e. the number of particles, was too small to obtain a more accurate value of CMC. This highlights an issue with using MD to predict CMC: careful consideration must be given to ensure that the system is large enough and that enough timesteps have been performed to ensure convergence, but increasing these parameters leads to an increase in computational cost.

The computational cost in increasing the system size can be mitigated using coarse-graining, which reduces the number of interactions that must be computed by grouping atoms into beads, as well as enabling larger time steps, meaning that the same timescale can be simulated with fewer rounds of calculation.[6] These beads must be modelled differently than individual atoms and there are a few approaches to this. Vishnyakov et al.[7] used dissipative particle dynamics (DPD) to model the CMCs of C_8E_8, dodecyldimethylamineoxide (DDAO) and N-decanoyl-N-methyl-D-glucamide (MEGA-10). They proposed a methodology for obtaining reasonable parameters to describe the bead interactions based on known infinite dilution coefficients, γ_∞. Their calculated CMC for C_8E_8 had an error relative to the average experimental value equal to $0.07\log\mu M$, for DDAO it was $0.06\log\mu M$, and for MEGA-10 it was $0.06\log\mu M$.

Clearly, this approach can produce extremely accurate results. However, it is limited to compounds that can be parameterised by a known γ_∞. Although this value can also be simulated, this must be done very carefully as the error in this value will be compounded.

Furthermore, the authors do not discuss parameterising ionic surfactants, for which more complex interactions must be considered. Despite the reduced cost relative to AA-MD models of the same system scales, CG-MD approaches are still one of the more expensive approaches to CMC prediction. But MD simulations give very deep insight into the processes occurring at a molecular level; its results yield a precise description of the arrangement of molecules in a system, their shape and aggregation number, which is a level of detail other approaches do not come close to achieving.

Another approach is to try to reduce the scale of the simulation by only considering the effect of individual molecules and local interactions. The conductor-like screening model (COSMO) provides a tool for decomposing the problem in this way, by treating a molecule as a cavity with a charged surface in a solvent that acts as a dielectric continuum.[8] The cavity's surface is described by the solvent-accessible surface of the molecule. The geometry of this surface combined with a segment-wise description of its polarising charges is called the COSMO-surface and it can be calculated using density functional theory (DFT).

COSMO for realistic solvation (COSMO-RS) adapts the model for more complex types of solvent.[9] Solvents only act like a dielectric continuum when they are capable of perfectly screening the COSMO-surface of a solute, i.e. every point on the molecule's COSMO surface is matched by a point of opposite polarity due to the configuration of the solvent molecules. Due to entropy considerations, as well as the fact that some solvent molecules do not have appropriate polar charges to match-up in this way, this is often a poor approximation. COSMO-RS uses statistical mechanics to determine the probability distribution describing how surface charge densities align between two molecules. This allows chemical potentials to be determined.

COSMO-RS has two major advantages over MD simulations. Firstly, it removes the need to study the configuration space of an entire ensemble of molecules; the problem is reduced just to local interactions. This means the computational cost does not scale with respect to the system size, but also that it can only consider homogeneous, unstructured systems.

Another advantage is that it naturally considers effects from dipole and quadrupole moments, which can be hard to capture using the classical forces that drive MD.[9]

Turchi et al.[10] 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 modelled 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 could then be determined by the equilibrium surfactant concentration in the aqueous phase. In actuality, micelles are structured and inhomogeneous; the authors argued that this approximation is more valid 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, containing primarily tail groups, which is particularly true as the interactions between head and tail groups become more unfavourable. This is the case for highly polar head groups: primarily for ionic surfactants.

The authors recommended applying both strategies and using the lower result as the CMC prediction. With the combined strategies, they attained an RMSE of 0.81 log µM on a dataset of 24 surfactants, containing a mix of ionic, nonionic and zwitterionic. Excluding the two worst predictions gives a much more favourable RMSE of 0.55 log µM. These results are relatively inaccurate, but it is impressive that the technique can be applied across all classes of surfactant, whilst requiring significantly lower computational cost than MD. Furthermore, the authors were able to determine which of the nonionic surfactants had a low propensity to form micelles by whether the predicted IFT between water and surfactant phases was large

at equilibrium, which is a testament to the model's interpretability.

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.[11] To compute these charge profiles, the layer's composition with respect to individual atoms of the surfactant must be known; this can be determined using MD. The combination of layer-wise atomic distributions and the COSMO-surface associated with each atom gives the layer's surface charge density profile. COSMOmic considers how the COSMO-surface of a surfactant would intersect with the layers' surface charge profiles when it is randomly positioned and oriented within the micelle, in order to determine the partition coefficient of inserting a surfactant molecule into the micelle.

Jakobtorweihen et al.[12] calculated CMCs using COSMOmic by first performing MD simulations to attain the layer-wise atomic distributions. As discussed above, MD by itself can calculate the CMC of a system, which might suggest that the COSMOmic step is redundant. However, one can expect the layer-wise atomic distributions to converge much more rapidly than equilibrium concentrations if the micelle is *pre-assembled*, meaning that the initial configuration for the simulation constitutes a guess for the micelle's structure. The authors then predicted the CMCs of several polyoxyethylene alcohols by determining the partition coefficients of inserting the respective surfactant monomer into a micelle.

Another consideration with this technique is that molecules may adopt several conformations, each with a unique COSMO-surface. Although it was found that the choice of conformer used to describe the micellar layer-wise charge density profiles has a negligible effect on the results,[13] the conformer used for the partitioning surfactant is important. With the best choice of conformer, the authors achieved an RMSE of $\sim 0.36 \log \mu M$ on predictions for C_iE_6 surfactants, where $i \in \{6, 8, 10, 12, 14, 16\}$. However, when considering surfactants with fixed tail length but varying head group size, $C_{10}E_j$, $j \in 4, 6, 8$, the authors could not identify a consistent conformer selection that would not yield significant outliers.

The technique is a promising compromise between the accuracy of the full-MD approach and the lower computational cost of COSMO-RS. However, to improve the reliability, a method for accurately identifying a good choice of conformer and the shape of the pre-assembled micelle must be realised.

Finally, COSMOplex is a recent extension of COSMOmic that removes the need to perform and initial MD simulation to determine the micellar structure.[14] Instead, it optimises a guess of the initial structure using a self-consistent approach by considering COSMOmic's predicted probability distribution of the partitioning surfactant's configuration within the micelle to iteratively yield new estimates for the layer-wise charge distributions. In this way, the authors predicted the CMCs of 10 nonionic surfactants with varied head and tail group chemistries, achieving an RMSE of 0.86 log µM. The authors note that these results are preliminary and stand to be improved.

The advantage of this technique is both a decrease in computational cost and improved parsimony; by eliminating the MD step, there is reduced complexity and the aforementioned difficulties with force field considerations are irrelevant. One of the difficulties with the technique is its sensitivity to the original guess for the atomic distributions.[14] Improving the guess or the optimisation technique to be less sensitive to local minima would therefore lead to more accurate predictions.

COSMO techniques are an interesting method to reduce the dimensionality of self-assembly structure studies whilst maintaining a solid grounding in theory, which allows a researcher to directly relate the results to scientific principles whilst understanding the limits of the approximations *a priori*. However, Herbert[15] have argued that the discussion around COSMO-RS indicates that there is not enough detail in the literature to allow for an open source implementation, meaning that many modern extensions are only available in the proprietary software package COSMOTHERM.[16] The closed-source nature of the technology is a barrier to scientific scrutiny and, crucially, improvement of the model.

Li et al.[17] applied a segment-based UNIQUAC model (s-UNIQUAC) and a SAFT equa-

tion of state to predict CMCs of linear polyoxyethylene alcohols by first deriving expressions for the activity coefficient of a surfactant in water. In the s-UNIQUAC model, a segment-based local-composition model was used, and the fugacity could then be approximated using the fitted interaction energies between the segments and water. In this case, the segments used were C_2H_4 and C_2H_4O. In the SAFT approach, the surfactant was treated as a chain of soft-sphere segments in order to first derive the Helmholtz energy of the solution and, from that, derive the fugacity. In this case, the segments used were CH_2/CH_3 (these were treated as the same segment) and C_2H_4O. The interaction energies of the segments were fitted, as well as parameters of a function describing the soft sphere diameter of a segment in a chain in terms of the chain length.

Cheng and Chen [18] compared the performance of these models on a larger dataset alongside three other models. Two of these were segment-based models: the polymer-NRTL model [17] and a UNIFAC model, [19] both of which were cited as inspirations for the s-UNIQUAC model. The authors also employed their own modified Aranovich and Donohue (m-AD) model. The m-AD model calculates the CMC as a mole fraction, x_S^L, approximating it as the reciprocal of the limiting value of the surfactant's activity coefficient in an aqueous solution, $\gamma_S^{L,\infty}$:

$$x_S^L = \frac{1}{\gamma_S^{L,\infty}} \tag{2}$$

The m-AD model considers the exchange equilibrium on a three-dimensional lattice of infinitely separated solvent and solute molecules in order to determine $\gamma_S^{L,\infty}$. Notably, the m-AD model is not a segment-based model; instead, the authors fitted an interchange energy, Δ, separately for each molecule. Of course, if a new parameter must be fitted for every molecule, a model has no predictive ability. Therefore, correlations were examined between Δ and other, readily calculated surfactant values: the Kier-Hall zero-order index (KH0) of the tail groups, which indicates and the total molecular energy of the surfactant.

Where data from the literature was available, the predictive performance 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 summarised in Table 1. The models all have a reasonably good accuracy, but the SAFT model in particular is excellent.

Table 1: Comparison of the RMSEs of selected models on polyoxyethylene alcohols. Data from Cheng and Chen[18].

Model	RMSE ($\log \mu M$)
p-NRTL	0.18
s-UNIQUAC	0.14
SAFT	**0.06**
UNIFAC	0.14
m-AD	0.11

Segment-based semi-empirical methods are therefore very promising for predicting CMCs within a class of surfactants. Their major drawback is that they are only applicable to molecules that can be decomposed into segments that have trained parameters. In addition, they must respect the limitations of the theories they are based upon. For example, it was described earlier how the SAFT model needs special treatment to be applied to this class of surfactants, requiring a function to describe a segment's soft sphere diameter with respect to the chain length. A more complex function would be required when branching is introduced, or when chains can be made up of combinations of different units, and it is difficult to make these adaptations.

Finally, purely empirical methods have a very heavy reliance on data abundance. Empirical methods offer a way of making predictions even when a unified theory that ties in the behaviour of several classes of molecules is not readily apparent, or else when the theory requires computationally expensive procedures to put into practice. However, without an underlying theory, the assumptions and therefore the limitations of the model are not well defined and it is possible for the model to 'learn' trends that contradict established scientific intuition.

The two fundamental differences between them are the choice of molecular descriptors,

the numerical features that form the basis set for the model inputs, and the functional form of the approximator that maps the descriptor to the property prediction.

Recently, an approach based on graph neural networks (GNNs) has produced highly accurate predictions whilst being applicable to nonionic, cationic, anionic and zwitterionic surfactants simultaneously.[20] 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.[21] Using such complex models, we also abandon the parsimony exhibited by Stauff-Klevens, and chemical insights can be much more difficult to derive. Furthermore, deep neural networks' ostensible 'universality' can be misleading: extrapolating the model's results to out-of-domain molecules (ones that are 'dissimilar' from the training data) will yield unreliable and potentially misleading predictions.

In this article, we develop two families of models of very different complexity: a linear model and a GNN. We evaluate the difference in performance and interpretability of the models. We also apply a technique for adding uncertainty quantification to the GNN, which can indicate whether a molecule is within the model's applicability domain and, therefore, whether a given prediction is reliable.

Method

Two datasets were used for training and testing:

The Qin dataset is a dataset of 202 surfactants, curated by a previous work[20] by accumulating results from several publications. To the authors' knowledge, it is currently the largest public dataset of CMCs for several classes of surfactant collected at standard conditions in an aqueous environment between 20 °C to 25 °C.

The NIST dataset is a dataset of 43 unique surfactants and their aqueous CMCs, extracted from the work of Mukerjee and Mysels[22]. For each surfactant, the mean of the experimental measurements between 20 °C to 25 °C and with no additives was used as

the target CMC value. These data were used exclusively for testing, to assess whether sampling bias affects the evaluated performance.

The Qin data[20] were split into training and test subsets to simulate the real-world scenario of using a model to make inferences about molecules for which no data is available. The training data were used to fit the models, whilst test data were 'locked away' until it had been decided that the model was optimised, and the performance metrics on the test data were used for evaluation. For some models, the training data were further split into optimisation and validation subsets; the optimisation data were used when calculating the loss function during model fitting. The validation data were used for on-the-fly evaluation of model performance during training.

To provide a consistent benchmark of model performance, the same data splits were used as Qin et al.[20]. The number of each class of surfactant in the train and test subsets of the data are shown in Table 2. Only the models trained on the Qin-All dataset were evaluated on the NIST dataset, as it contained ionic compounds.

Table 2: The number of each class of surfactant contained in the train/test subsets of the CMC datasets.

Data subset		Number of			
Dataset	Train/test	Nonionics	Anionics	Cationics	Zwitterionics
Qin-All	Train	110	30	31	9
	Test	12	4	4	2
Qin-Nonionics	Train	110			
	Test	12			
NIST	Train				
	Test	12	23	6	2

A QSPR pipeline requires choosing two essential functions: a representation function, whose parameters are defined before training the model, and a mathematical form that maps this representation to a prediction. The processes by which these functions are developed are called *feature engineering* and *model selection*, respectively.

Feature engineering

The ideal molecular representation depends on the task at hand. Ideally, it should be compact but complete;[23,24] 'as simple as possible, but not simpler.' To that end, the representation should contain enough information to distinguish between isomers that 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,[25] but this is redundant in cases where the training data spans a very limited range of states.

Extended-connectivity fingerprints

In this approach, the molecule is split into atomic environments up to a given radius, r: each environment is centred on an atom and extends r steps along connecting bonds. Effectively, we discard the categorical encoding in favour of introducing more continuous, count-based features, like n_c. The set of all environments in the training data up to radius r is extracted. The resulting feature vector, $\vec{c}$ for a molecule is described by

$$c_m = \text{Count}(\mathcal{E}_m), \tag{3}$$

where $\mathcal{E}_m$ represents the m^{th} atomic environment.

Now, a change in headgroup composition is reflected in a change in subgraph counts, and provided the new subgraph exists in our 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. This type of representation is called an *extended-connectivity fingerprint* (ECFP).[26]

ECFPs are similar to a segment-based approach; however, unlike segments or groups, subgraphs can overlap. As discussed above, a group contribution approach requires that a canonical 'priority' of the groups be defined prior to featurising molecules. By using ECFPs,

the manual identification of important groups and their priorities are skipped; feature importance determination is instead delegated to the model.

However, these fingerprints do not necessarily distinguish between all positional or chain isomers, particularly with smaller values of r, nor are stereoisomers treated differently. Another potential disadvantage is that 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 model fitting and interpretation.

Molecular graph representation

Both of the approaches described so far rely on molecular feature vectors that cannot describe the molecule's topology. A molecular graph is a structure that achieves this, and it is a popular choice for cheminformatics as well as visualisation of molecular structure. In this approach, 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, hybridisation state, charge, etc. These feature vectors are concatenated into a node feature matrix, $\mathbf{V}$. The graph's structure is then defined by a binary adjacency matrix, $\mathbf{A}$:

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

Molecular graphs are perhaps the most natural representations to visualise; see Figure 1. This is an exact description of the molecule's topology that enables an atomistic machine learning approach.

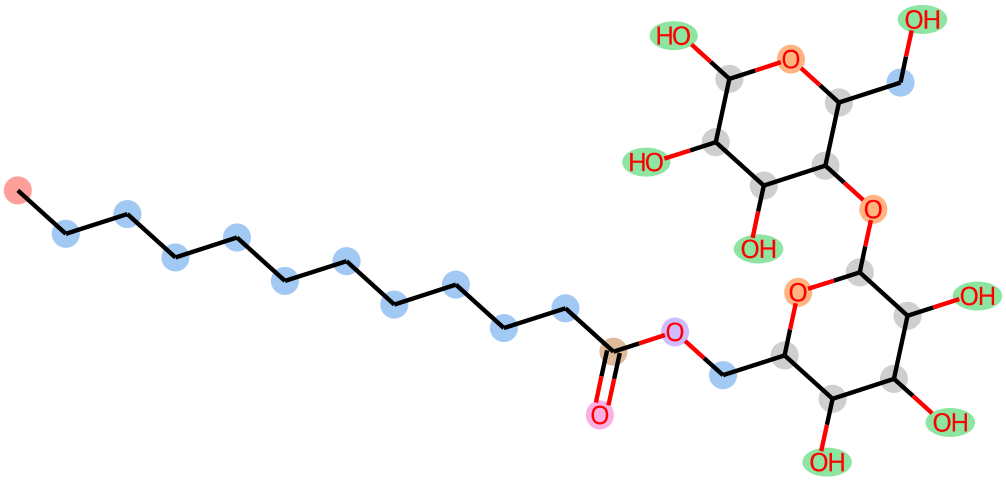


Figure 1: A molecular graph of 6-*O*-dodecanoyl-maltose. Atoms are highlighted based on their feature vectors, $\vec{v}_i$, so that equal feature vectors have the same colour.

Model selection

ECFP model

Based on the prior knowledge encoded in Equation 1, it is reasonable to assume 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 Equation 3:

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

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; a naïve fit using ordinary least squares (OLS) would likely produce poor results. To that end, a process of *feature selection* was applied, whereby a subset of the atomic environments were selected for use in the model. There are several approaches to feature selection;[27] here, we chose an approach based on *regularisation*.

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 in this paper are ℓ_1 and ℓ_2 regularisation, which

correspond to the inclusion of ℓ_1 and ℓ_2 norms, respectively. By combining ℓ_1 regularisation with the least squares regressor, we obtain the least absolute shrinkage and selection operator (LASSO):[28]

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

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 is a user-defined hyperparameter describing the degree of regularisation ($\alpha \geq 0$). $\mathbf{C}$ are the standardised training data feature vectors, $\{\vec{c}'_n \,|\, 1 \leq n \leq n_{\text{samples}}\}$, stacked row-wise into a matrix. Standardising 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 standardisation is necessary to ensure that the regularisation term is not dominated by environments with high variance and that it accounts for common and uncommon environments alike.

By imposing the ℓ_1 penalty, the model is biased towards learning a *sparse* weight vector: many of its elements will be negligible. The corresponding features can be removed from the representation.

However, LASSO has two major flaws that make it inappropriate for the task of ECFP regression:

- The number of unique atomic environments is greater than the size of the training data, n_{samples}, but LASSO will select at most n_{samples} features.[29] Therefore, some important environments may still be excluded.

- LASSO tends to select only one of a group of highly correlated variables when there is

no reason why that particular one should be prioritised.[30] This is undesirable because we want to include both large and small atomic environments, despite their large correlation, and we might be misled into thinking that some highly correlated environments have no effect on CMC.

ElasticNet addresses these issues by imposing an additional ℓ_2 penalty:[30]

$$\min_{\vec{w}} \frac{1}{2n_{\text{samples}}} \left\| \mathbf{C}\vec{w} + \vec{b} - \vec{y} \right\|_2^2 + \alpha\rho \left\| \vec{w} \right\|_1 + \frac{\alpha(1-\rho)}{2} \left\| \vec{w} \right\|_2^2, \tag{8}$$

where ρ is a user-defined hyperparameter controlling the proportions of the regularisation terms, $0 < \rho < 1$. This removes the hard limit on the number of features that can be selected and also exhibits the 'grouping effect', whereby features with high correlation tend to be assigned similar weights.

Because of these advantages, an ElasticNet linear regression model was applied to select the most important ECFP features for predicting $\log X_{cmc}$. Both hyperparameters, α and ρ must be defined when training the model. In order to select the best values, k-fold cross-validation was employed: the training data were partitioned into k subsets of roughly equal size. k models were trained using $k-1$ subsets, and the final subset was used for validation. The average mean-squared error of the k models, evaluated on their respective validation subsets, is the model's final score. This routine, with $k = 5$, was applied for a range of α and ρ combinations, and the lowest-scoring combination was used. The hyperparameter search space is defined in the Supplementary 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 regularisation so that all of the weights are non-negligible, but we still address the issue of multicollinearity:[31]

$$\min_{\vec{w}} \left\| \mathbf{C}\vec{w} + \vec{b} - \vec{y} \right\|_2^2 + \alpha \left\| \vec{w} \right\|_2^2. \tag{9}$$

A similar cross-validation method to the one described above was used to determine the

best α parameter, but using $k = n_{\text{samples}} - 1$. 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, likely due to using a larger number of folds when determining the best value for α. Both models were implemented using scikit-learn.[32]

Molecular graph model

The basic topology of the graph neural network (GNN) used in this work was identical to the one used by Qin et al.[20]. That is, 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.[33] Layer l computes a new node feature matrix, $\mathbf{V}^{(l)}$, based on the adjacency matrix, $\mathbf{A}$:

$$\mathbf{V}^{(l)} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}\mathbf{V}^{(l-1)}\mathbf{W}^{(l)} + \mathbf{b}^{(l)}. \tag{10}$$

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

$$D_{ii} = \sum_j A_{ij}, \tag{11}$$

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

$\mathbf{V}^{(1)}$, therefore, encodes not only information about the atom itself but its bonded neighbours. This information is used in the subsequent graph convolution so that $\mathbf{V}^{(2)}$ encodes information about the 2^{nd} order neighbourhood, *et cetera*. The number of graphs 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

characterised 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 the explicit topological information. Several choices of pooling function were trialled:

Mean pooling was used in the previous work. It computes the average over all 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:[34]

$$\vec{v}^{(p)} = \sum_i^N \sigma\left(\mathbf{W}_1\vec{v}_i^{(L)} + \vec{b}_1\right) \odot \left(\mathbf{W}_2\vec{v}_i^{(L)} + \vec{b}_2\right), \tag{12}$$

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:

$$\mathbf{X} = \text{softmax}(\mathbf{V}^{(L)}\vec{w}), \tag{13}$$

$$\vec{v}^{(p)} = \sum_i^N \mathbf{X}_i \cdot \vec{v}_i^{(L)}, \tag{14}$$

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 optimised representation is a feature of neural networks that happens implicitly during training, called *representation learning*.[35] The final step is a readout neural network: a multi-layer perceptron which 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, $\vec{v}^{(l)}$:

$$\vec{v}^{(l)} = \mathbf{W}^{(l)}\vec{v}^{(l-1)} + \vec{b}^{(l)}, \tag{15}$$

The full network's architecture is illustrated in Figure 2. The model was implemented using the open-source library Spektral.[36]

A neural network's topology describes the types of layers used, i.e. their functional form and the connection between them. Layers that are parameterised by a weight matrix, $\mathbf{W}$, may have different 'sizes', meaning that the dimensionality of their output is arbitrary and can be adjusted by changing the dimensions of $\mathbf{W}$. 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.

Optimising GNN hyperparameters

The Hyperband approach,[37] implemented in Keras Tuner,[38] 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 trials 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, whilst 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

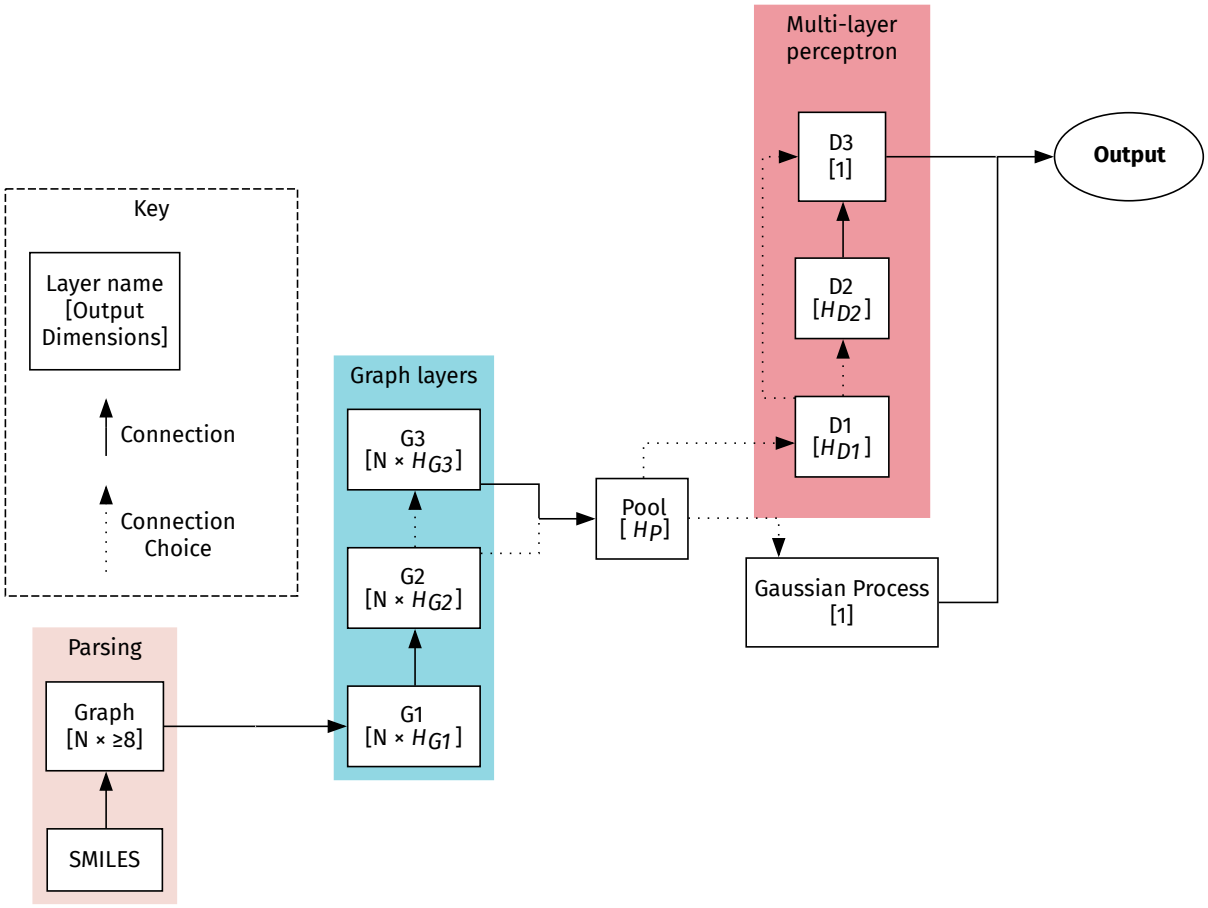


Figure 2: Schematic of the neural network architecture. Here, N represents the number of constituent atoms/ions in the input molecule and H represents a hyperparameter. The size of the pooling layer output, H_P, is only independent in the case of a gated attention-pooling layer. Otherwise, it is equal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

iterates 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 was partitioned into an optimisation subset and a validation subset in a ratio of 9:1. The trials were fit to the optimisation subset and evaluated based on the RMSE of their predictions on the validation subset.

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.[39]. The model first computes the latent representation vector, $\vec{v}^{(p)}$, of an input molecule using a trained GNN. $\vec{v}^{(p)}$ is then standardised, similar to Equation 7:

$$v_n^{(p)\prime} = \frac{v_n^{(p)} - u_n}{s_n}, \tag{16}$$

but in this case, the standardisation applies across each latent feature, n. Again, u_n and s_n were determined from the training molecules' latent representations.

The standardised 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, $\log X_{cmc} \sim \mathcal{N}(\mu, \sigma)$.

In this work, the GPs were defined using a Matérn kernel with parameter 1/2. Furthermore, the multi-layer perceptron component of the GNN used to calculate $\vec{v}^{(p)}$ was employed as the GP's mean function. The kernel parameters were optimised with an Adam optimiser.[40] The same optimisation/validation splitting was used as for the GNN hyperparameter search and training was stopped after 1000 iterations without improvement in the validation predictions' RMSE. This implementation was based on GPFlow.[41]

Results

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 dataset is similar at approximately 74 %, and so is the ratio of the initial number of features to the number of selected features, 31 % to 33 %. The number of features is large compared to many of the empirical models described above but not the number of graph network parameters. Furthermore, this model also aims to cover a large part of chemical space, so a large number of parameters is to be expected.

Table 3: The number of atomic environments at each stage of the ECFP feature selection process.

	Number of training data atomic environments		
Dataset	Initially	Found in multiple molecules	With non-negligible weight
Qin-Nonionics	260	201	81
Qin-All	410	302	134

Hyperband tuning

725 trials were conducted for each of the Qin training datasets. The best hyperparameters discovered on each set are described in Table 4.

Model performance

The performances of all the trained models on the test datasets are reported in Table 5. All of the models outperformed those of the previous work. For every task, the most accurate model was either the GNN or the combined GNN with the GP (GNN/GP). However, the linear model's performance is surprisingly good, considering its relative simplicity, faster optimisation and the far smaller number of parameters it constitutes.

Table 4: The best hyperparameters discovered during searching. 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 weren't 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.

	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}	–	–

Table 5: Test dataset evaluation results for the models trained in this work versus those of the previous work. The best RMSE for each task is emboldened.

	Test RMSE (log µM)		
Model	Qin-Nonionics	Qin-All	NIST
Previous work[20]	0.23	0.30	–
ECFP	0.19	0.26	1.59
GNN	**0.15**	0.29	**1.35**
GNN/GP	1.38	**0.24**	1.45

The performance of the NIST data is significantly worse than the test data performance of every model. This suggests that the NIST data molecules are outside of the applicability domain of the models.

Finally, 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.

Uncertainty quantification

However, the RMSE 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:

$$\mathrm{NLL} = -\sum_n \log p_n(\hat{y}_n), \tag{17}$$

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 supplementary information for comparison against future work.

To assess the models' quality individually, the predictions can be visualised against the true CMCs in a parity plot; see Figure 3a. Alternatively, a calibration plot can be used, which compares the distribution of the residuals against the expected distribution given the models' predicted normal distributions. The expected distribution simulates what would be observed if the residuals were drawn from the distributions predicted by the models. Deviations from this distribution indicate whether the model was over- or underconfident

(c.f. Tran et al.[39]). These 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. There is a spike in the number of observed residuals that are close to the centre of the distribution. The corresponding parity plot shows that, nevertheless, the predicted uncertainties were relatively small. The NIST data calibration curve shows remarkably good agreement with the ideal distribution, except at the top end, which reflects the tendency of the model to underestimate the CMCs of some of the molecules. The relatively poor RMSE on the NIST data is somewhat ameliorated by the quality of the uncertainty estimates.

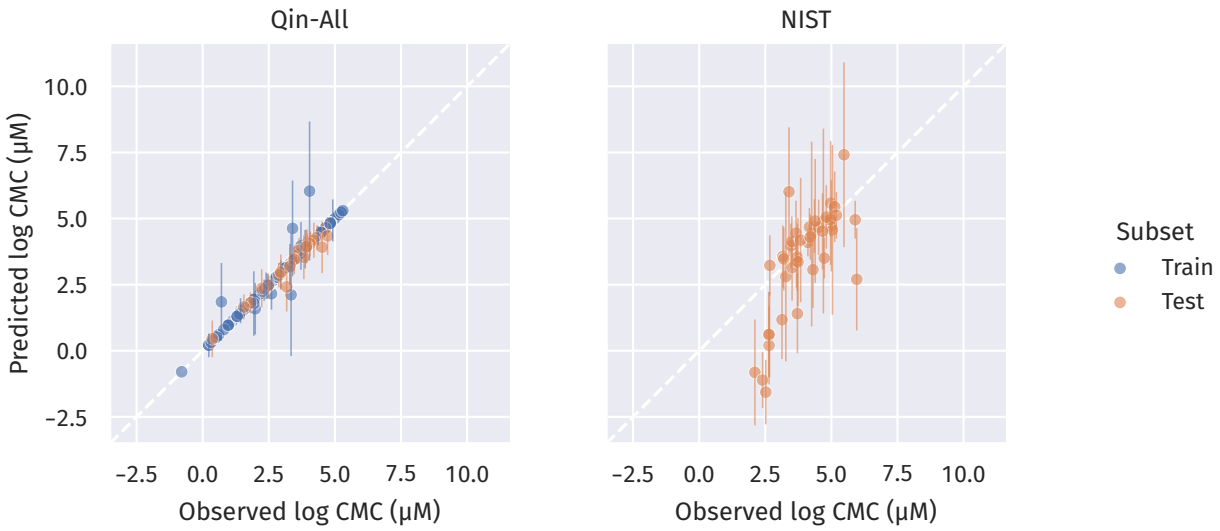
Future efforts to improve this type of model may consider incorporating another term in the loss function for the GNN that explicitly biases the model towards learning a form of $\vec{v}^{(p)}$ that captures this similarity. Alternatively, a variational Gaussian process could be used, which approximates the Gaussian process using a fixed-size set of 'pseudo-points';[42] this would enable the entire GNN/GP model to be trained at once using backpropagation.[43]

ECFP interpretations

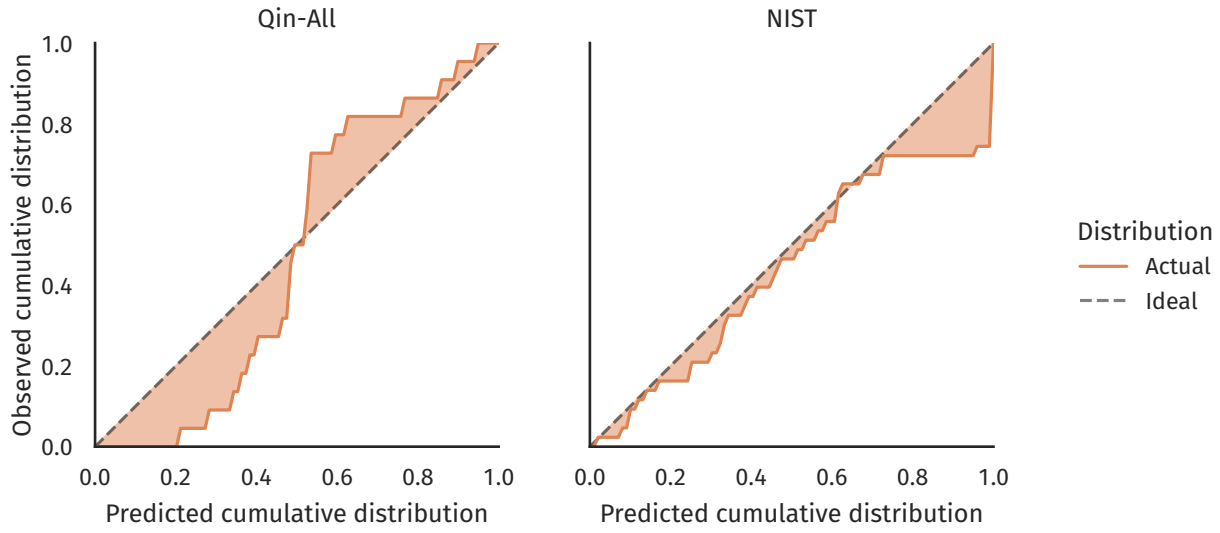
TODO: find out the hybridization states of the atoms in these fingerprints so that we can clarify the environments more:

- 4201881788
- 1435798937
- 3833245231
- 3204830367

The weights of the ECFP models are coefficients corresponding to the scaled counts of the selected atomic environments. Referring to Equations 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)



(b)

Figure 3: (a) Parity plots of the GNN/GP model's predicted CMCs and 95 % confidence intervals for the Qin-All and NIST datasets and (b) corresponding calibration plots for the test data predictions.

A more readily interpreted value can be achieved by rescaling the coefficient, w_m, for an environment:

$$w'_m = \frac{w_m(1 - u_m)}{s_m}, \tag{18}$$

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 contain 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 of larger environments that envelope smaller ones. The largest scaled weights for the two ECFP models are given in Table 6.

Table 6: The atomic environments with the greatest importance to 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	$(CH_2)_2CH$	−0.29
Br^-	0.29	C	−0.25
$(CH_2)_2CH$	−0.27	$C(CH(OH))_3CH$	−0.19
CH_2	−0.23	$CH_2(CH(OH))_3CH$	0.14
O	0.18	$CH(CH(OH))_2CH_2OH$	−0.12
OH	−0.17	$CH_2CH(O)(CH(OH))_2CH_3$	0.09
$O(CH_2)_2OH$	−0.14	CH_3	−0.06
$CH_2O(CH_2)_2OH$	−0.14	$CH(CH(OH))_3CH$	0.05

Both models agree that alkyl chain environments constitute the top two most important contributors to 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 distinguishes 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.

TODO: Discuss NIST data and applicability domain.

Conclusion

Empirical models were applied to predict CMCs from two datasets. One dataset 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 NIST dataset was collected from a different source and contained some molecules with very different chemistries than the above.

A linear model based on ECFPs demonstrated remarkably good performance, improving on a previous work[20] that applied a more complex GNN model, despite using a smaller number of parameters and having a much faster optimisation time. A new model was presented that improved the architecture of previous work's GNN and was capable of obtaining better performances than the ECFP model on the Qin-Nonionics task and demonstrated a better ability to generalise to the NIST dataset.

Finally, 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 appeared to fail when applied to the Qin-Nonionics task, it yielded the best predictive performance of all of the models for the Qin-All task, as well as providing a good quality of uncertainty estimates allows researchers to gauge their confidence in the model's predictions.

TODO: Write about applicability domain.

Supporting Information Available

Source code for featurisation and model training, graph neural network logs and metrics for hyperparameter optimisation and final training, and individual model predictions.

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. *Journal of the American Oil Chemists Society* **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. *The Journal of Chemical Physics* **1990**, *92*, 3710–3724

.

(4) de Miguel, R.; Rubí, J. M. Gibbs Thermodynamics and Surface Properties at the Nanoscale. *The Journal of Chemical Physics* **2021**, *155*, 221101

.

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

.

(6) Fitzgerald, G.; DeJoannis, J.; Meunier, M. In *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

.

(7) Vishnyakov, A.; Lee, M.-T.; Neimark, A. V. Prediction of the Critical Micelle Concentration of Nonionic Surfactants by Dissipative Particle Dynamics Simulations. *The Journal of Physical Chemistry Letters* **2013**, *4*, 797–802

.

(8) Klamt, A.; Schüürmann, G. COSMO: A New Approach to Dielectric Screening in Solvents with Explicit Expressions for the Screening Energy and Its Gradient. *Journal of the Chemical Society, Perkin Transactions 2* **1993**, *0*, 799–805

.

(9) Klamt, A.; Eckert, F.; Arlt, W. COSMO-RS: An Alternative to Simulation for Calculating Thermodynamic Properties of Liquid Mixtures. *Annual Review of Chemical and Biomolecular Engineering* **2010**, *1*, 101–122

.

(10) Turchi, M.; Karcz, A. P.; Andersson, M. P. First-Principles Prediction of Critical Micellar Concentrations for Ionic and Nonionic Surfactants. *Journal of Colloid and Interface Science* **2022**, *606*, 618–627

.

(11) 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. *The Journal of Physical Chemistry B* **2008**, *112*, 12148–12157

.

(12) Jakobtorweihen, S.; Yordanova, D.; Smirnova, I. Predicting Critical Micelle Concentrations with Molecular Dynamics Simulations and COSMOmic. *Chemie Ingenieur Technik* **2017**, *89*, 1288–1296

.

(13) Jakobtorweihen, S.; Ingram, T.; Smirnova, I. Combination of COSMOmic and Molecular Dynamics Simulations for the Calculation of Membrane–Water Partition Coefficients. *Journal of Computational Chemistry* **2013**, *34*, 1332–1340

.

(14) 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. *Physical Chemistry Chemical Physics* **2019**, *21*, 9225–9238

.

(15) Herbert, J. M. Dielectric Continuum Methods for Quantum Chemistry. *WIREs Computational Molecular Science* **2021**, *11*, e1519

.

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

.

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

.

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

.

(19) 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 Non-aqueous Solvents with UNIFAC. *Industrial & Engineering Chemistry Research* **2001**, *40*, 2362–2366

.

(20) Qin, S.; Jin, T.; Van Lehn, R. C.; Zavala, V. M. Predicting Critical Micelle Concentrations for Surfactants Using Graph Convolutional Neural Networks. *The Journal of Physical Chemistry B* **2021**, *125*, 10610–10620

.

(21) Bejani, M. M.; Ghatee, M. A Systematic Review on Overfitting Control in Shallow and Deep Neural Networks. *Artificial Intelligence Review* **2021**, *54*, 6391–6438

.

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

.

(23) Faber, F.; Lindmaa, A.; von Lilienfeld, O. A.; Armiento, R. Crystal Structure Representations for Machine Learning Models of Formation Energies. *International Journal of Quantum Chemistry* **2015**, *115*, 1094–1101

.

(24) 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. *Computer Physics Communications* **2020**, *247*, 106949

.

(25) Chen, C.; Ye, W.; Zuo, Y.; Zheng, C.; Ong, S. P. Graph Networks as a Universal Machine Learning Framework for Molecules and Crystals. *Chemistry of Materials* **2019**, *31*, 3564–3572

.

(26) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling* **2010**, *50*, 742–754

.

(27) Li, J.; Cheng, K.; Wang, S.; Morstatter, F.; Trevino, R. P.; Tang, J.; Liu, H. Feature Selection: A Data Perspective. *ACM Computing Surveys* **2017**, *50*, 94:1–94:45

.

(28) Tibshirani, R. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* **1996**, *58*, 267–288

.

(29) Efron, B.; Hastie, T.; Johnstone, I.; Tibshirani, R. Least Angle Regression. *The Annals of Statistics* **2004**, *32*, 407–499

.

(30) Zou, H.; Hastie, T. Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **2005**, *67*, 301–320

.

(31) McDonald, G. C. Ridge Regression. *WIREs Computational Statistics* **2009**, *1*, 93–100

.

(32) Pedregosa, F. et al. Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830

.

(33) Kipf, T. N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. 2017

.

(34) Li, Y.; Tarlow, D.; Brockschmidt, M.; Zemel, R. Gated Graph Sequence Neural Networks. 2017

.

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

.

(36) Grattarola, D.; Alippi, C. Graph Neural Networks in TensorFlow and Keras with Spektral. 2020

.

(37) Li, L.; Jamieson, K.; DeSalvo, G.; Rostamizadeh, A.; Talwalkar, A. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. *Journal of Machine Learning Research* **2018**, *18*, 1–52

.

(38) Chollet, F., et al. Keras. 2015

.

(39) Tran, K.; Neiswanger, W.; Yoon, J.; Zhang, Q.; Xing, E.; Ulissi, Z. W. Methods for Comparing Uncertainty Quantifications for Material Property Predictions. *Machine Learning: Science and Technology* **2020**, *1*, 025006

.

(40) Kingma, D. P.; Ba, J. Adam: A Method for Stochastic Optimization. 2017

.

(41) Matthews, A. G. d. G.; van der Wilk, M.; Nickson, T.; Fujii, Keisuke.; Boukouvalas, A.; León-Villagrá, P.; Ghahramani, Z.; Hensman, J. GPflow: A Gaussian Process Library Using TensorFlow. *Journal of Machine Learning Research* **2017**, *18*, 1–6

.

(42) Hensman, J.; Fusi, N.; Lawrence, N. D. Gaussian Processes for Big Data. 2013

.

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

.

TOC Graphic

I'm looking for ideas for the TOC graphical entry – common practice seems to be to draw a really abstract picture of a molecule going through a network, but that doesn't seem like the best thing to do here because I'm specifically comparing against other models.