Digital Discovery

PAPER

Cite this: *Digital Discovery*, 2025, **4**, 424

Predicting homopolymer and copolymer solubility through machine learning†

Christopher D. Stubbs, [a] Yeonjoon Kim, [b] Ethan C. Quinn, [a] Raúl Pérez-Soto, [a] Eugene Y.-X. Chen [*a] and Seonah Kim [*a]

Received 9th September 2024
Accepted 12th December 2024

DOI: 10.1039/d4dd00290c

rsc.li/digitaldiscovery

Polymer solubility has applications in many important and diverse fields, including microprocessor fabrication, environmental conservation, paint formulation, and drug delivery, but it remains under-explored compared to its relative importance. This can be seen in the relative scarcity of solvent-based systems for recycling plastics, despite a need for efficient and selective methods amid the looming plastics and climate crises. Towards this need for better predictive tools, this work examines the use of classical and deep machine learning (ML) models for predicting categorical solubility in homopolymers and copolymers, with model architectures including random forest (RF), decision tree (DT), naive Bayes, AdaBoost, and graph neural networks (GNNs). We achieve high accuracy for both our homopolymer (82%, RF) and copolymer models (92%, RF) on unseen polymer–solvent systems in our 5-fold cross-validation studies. The relevance and applicability of our homopolymer models are then verified through in-house experiments examining the solubility of common commercial plastics, followed by an explainable AI (XAI) analysis using Shapley Additive Explanations (SHAP), which explores the relative contribution of each feature toward model predictions. We then apply our homopolymer solubility prediction model to remove unwanted or hazardous additives in polyethylene (PE) and polystyrene (PS) waste. This work demonstrates the validity/feasibility of using ML to predict homopolymer solubility, provides novel ML models for the prediction of copolymer solubility, and explains homopolymer model predictions before applying the explained model to a globally relevant waste challenge.

Introduction

As one of the cornerstones of modern materials, polymers have become near ubiquitous due to their low cost and diverse physical and chemical properties. Polymers are used in a highly diverse range of applications and fields, from children's toys to spacecraft. Polymer solubility is significant in these applications, impacting the safety, durability, and processing of many polymers. Furthermore, polymer solubility is especially critical in drug release, nanofiltration, and polymerization reaction design.[1,2] In addition to these important applications, one application that particularly stands out in its global relevance is solvent-based plastic recycling, which can cost-effectively address the environmental harm posed by untreated plastic waste.[3–11] For solvent recycling and the applications mentioned above, one must understand what solvents dissolve which polymers – which motivates a discussion of polymer solubility.

Polymer solubility can be differentiated from small molecule solubility in many ways, significantly impacting its prediction and metrics.[12,13] One such difference is the timescale/speed of solute diffusion: because polymer chains are large and susceptible to entanglement, the diffusion of the polymer solute is often far slower than the diffusion of the solvent – which is not universally true for small molecules. Another important distinction between small molecule and polymer solubility is the number of phases typically formed – while small molecules often form 2–3 phases during dissolution, polymers generally form between 4 and 6 phases – which can include a pure polymer layer, an infiltration layer, a gel layer, and a pure solvent layer; this adds significant complexity to theoretical models of solubility.[12,13] A third distinction between small molecule and polymer solubility can be seen in the solubility metrics used. In general, because polymers are statistical ensembles of macromolecules, they have qualitatively different dissolution from small molecules; this makes quantifying their degree of solubility difficult as most measures of solubility are not well-defined.[12,13] There are three typical measures of polymer solubility: the Flory–Huggins χ parameter, the Hildebrand solubility parameter, and the Hansen solubility parameter.[13] Such parameters aim to approximate the change in free energy associated with polymer dissolution, but their methods vary.

[a] *Department of Chemistry, Colorado State University, Fort Collins, CO 80523-1872, USA. E-mail: eugene.chen@colostate.edu; seonah.kim@colostate.edu*

[b] *Department of Chemistry, Pukyong National University, Busan, Republic of Korea*

† Electronic supplementary information (ESI) available. See DOI: https://doi.org/10.1039/d4dd00290c

The Flory–Huggins χ parameter is most commonly defined using a lattice model of polymer solubility, where z solvents (the coordination number) surround each polymer segment, interacting with the polymer to cause an energy change of $\Delta\varepsilon$ at temperature T (eqn (1A)).[12] In contrast, the Hildebrand solubility parameter (δ) describes polymer solubility using the cohesive energy density (CED), which is the energy required to break all intermolecular interactions per unit volume; in practice, the CED is typically approximated using the heat of vaporization (ΔH_{vap}) and reference volume (V) at temperature T (eqn (1B)).[13] The Hansen solubility parameters separate this CED into dispersive (δ_{D}), polar (δ_{P}), and hydrogen bonding (δ_{H}) components (eqn (1C)).[13]

Eqn (1): definitions of the most common measures of polymer solubility.

$$\begin{aligned}
&\textbf{A: Flory}-\textbf{Huggins }\boldsymbol{\chi}\\
&\chi \equiv \frac{z \times \Delta\varepsilon}{k_{\mathrm{B}}T}\\
&\textbf{B: Hildebrand solubility parameter}\\
&\delta \equiv \sqrt{\mathrm{CED}} = \sqrt{\frac{\Delta E_{\mathrm{coh}}}{V}} \approx \sqrt{\frac{\Delta H_{\mathrm{vap}} - RT}{V}}\\
&\textbf{C: Hansen solubility parameter}\\
&\delta \equiv \sqrt{\mathrm{CED}} = \sqrt{\frac{\Delta E_{\mathrm{coh}}}{V}} \approx \sqrt{{\delta_{\mathrm{D}}}^2 + {\delta_{\mathrm{P}}}^2 + {\delta_{\mathrm{H}}}^2}
\end{aligned} \tag{1}$$

While all three parameters have found successful applications,[2,14,15] each has its shortcomings. The frequently used lattice model of the χ parameter does not account for variations in chain packing across a polymer sample; while alternative formulations exist, there does not appear to be widespread consensus on the most broadly applicable definition.[12] Furthermore, the CED in the Hildebrand and Hansen solubility parameters is often defined using the heat of vaporization – which is not defined for all polymers due to sample decomposition before vaporization.[13] To avoid these shortcomings and others, we can consider whether a specific polymer is soluble or insoluble in a given solvent. This approach, termed binary solubility labels, is informative and succinct as it dramatically reduces the theoretical complexity of predicting polymer solubility. Despite this reduction in complexity, there are still significant computational challenges in predicting polymer solubility – primarily due to the size of polymer systems. Molecular dynamics (MD) and density functional theory (DFT) calculations are generally too costly to apply to polymer systems of reasonable chain lengths, and performing hundreds of these calculations would be prohibitively expensive. Furthermore, accounting for parameters such as polymer morphology or crystallinity adds even more complexity – motivating the search for a low-cost, high-accuracy approach to predicting polymer solubility.

To fulfill this need, we use machine learning (ML) – a broad class of statistical algorithms designed to make predictions from an input database. ML has received significant attention in recent years due to its speed and accuracy compared to *ab initio* or semi-empirical methods.[16–19] This makes ML particularly well suited for systems where the methods mentioned above are too costly or insufficiently describe the system. In particular, ML is well suited for polymer solubility towards solvent recycling, given the broad scope of polymers that may exist in current and future waste streams. Chemical information about each polymer/solvent is represented through one or more chemical descriptors for each system. Chemical descriptors indirectly describe a molecule by tabulating electronic, steric, and structure-based properties, using these properties to infer a prediction target (in this case, the binary solubility label). To represent the polymer dissolution process, we argue that one should account for both polymer–solvent energetic interactions (*e.g.*, number of aromatic rings) as well as for steric interactions and diffusion (*e.g.*, Hall–Kier connectivity and shape indices[20]), both of which play a crucial role in polymer solubility.[13]

There have been several reported works on predicting polymer solubility using ML, which is unsurprising given its importance. Most such reports have focused on the previously described Flory–Huggins χ and Hildebrand/Hansen solubility parameters.[21–23] In addition to the theoretical problems outlined above regarding polymer vaporization and lattice theory, previous literature has found that data on the Hildebrand and Hansen parameters is scarce and that models built upon them generally show subpar performance of 60–75% accuracy.[24] Others have considered binary solubility labels and seen improved success; Ramprasad and coworkers published one example in 2020, where the authors used a deep neural network to predict binary solubility labels at relatively high accuracy.[25] This work examined limited solvents (24) and architectures (1) for polymer solubility, limiting the model's predictive ability for unseen or novel solvents. A follow-up to this work by Kern *et al.* predicted binary solubility labels using a RF model alongside an improved version of the previous neural network.[26] However, solvent scope (51) and ML architecture scope (2) were still limited.[26]

While these works represent informative and significant advances in the field and have informed parts of this work, several areas are not covered, which we address herein. These include homopolymer and copolymer solubility predictions, comparisons of multiple ML architectures, and a thorough analysis of the relationship between descriptor choice and model performance. In particular, the absence of copolymer solubility predictions in literature represents a significant disconnect between state-of-the-art computational polymer chemistry and experimental polymer chemistry, where copolymers play an important role in developing new materials. As polymers with two or more repeat units, copolymers have risen to this prominence due to their stoichiometry-modulated properties and their ability to rapidly self-assemble into sophisticated nanostructures with applications in energy storage and light responsive materials.[27–29] While ML predictions have been reported for the thermal,[30–32] mechanical,[33,34] optical[35] and morphological[36–38] properties of copolymers, so far there have been no reports predicting copolymer solubility for a broad range of copolymers.[39,40] We set out to remedy this gap in the literature, among others, by predicting copolymer solubility for a diverse set of copolymers and associated solvents through both classical and deep ML models.

This report uses classical and graph-based ML models to predict homopolymer and copolymer solubility at high accuracy over various descriptors. We performed in-house experiments to validate our homopolymer models and utilized an explainable artificial intelligence (XAI) analysis to explain model performance. We then predict selective solvents for additive removal in polyethylene and polystyrene and propose multiple solvent systems for efficient additive removal. Our models make robust and accurate predictions from only user-supplied chemical names and cover various solvents and polymers, making solubility predictions rapid, accessible, and relevant.

Methods

Polymer solubility database

An initial literature database of binary solubility labels was constructed from a previously published polymer handbook by Brandrup *et al.*,[41] with the database undergoing significant preprocessing before any models were trained. This preprocessing mapped each polymer and solvent to a text representation (SMILES), with any unresolvable or ill-defined molecules discarded. Polymers were mapped to their presumed repeat unit based on their polymer name for tractability and simplicity. Any data points which specified partial solubility, cosolvents, elevated temperatures, polymer impurities/dopants, polymer crystallinity, polymer tacticity, molecular weight, or polymer morphology were discarded. These categories either did not contain sufficient data to make reasonable conclusions or were outside the scope of this work (*e.g.*, partial solubility). Additional details on the definition of solubility used can be found in the ESI† ("Definition of Binary Solubility"). Following preprocessing, the initial database was separated into two input databases: a more extensive homopolymer and a smaller copolymer (see Table 1). The homopolymer database consisted of 1818 homopolymer–solvent pairs with labels of soluble/insoluble, including 175 unique solvents.

In contrast, the copolymer database had only 270 copolymer–solvent pairs, which is lower than the homopolymer database due to the relative scarcity of copolymer solubility data (Table 1). It should be noted that here, we define good solvents as those in which a polymer chain expands relative to the pure polymer (soluble), whereas in bad solvents, the polymer chain contracts (insoluble). This definition is primarily theoretical and relates to the energetics of solvation rather than the dynamics such as diffusion, but we accept this compromise to increase the tractability of our approach.

Descriptors

While our classical and graph-based models predict from input descriptors, the specific descriptors used differed significantly. Here, 'classical model' refers to any non-neural network-based ML model whereas 'graph-based model' refers to our message passing neural network (MPNN) ML model. Seven different descriptor groups were evaluated for the trained classical ML models, representing two broad categories: molecular and fingerprint descriptors (Fig. 1). The descriptor groups used were chosen from chemical intuition for the dissolution process and using automated processing criteria (any descriptors that had a value of zero for 98–99% zero across the entire database were discarded).

For example, the number of hydrogen bond acceptors was chosen as this closely relates to solubility; similarly, the Morgan and RDKit fingerprints were selected to represent structural information directly, which is also closely related to solubility. For every polymer–solvent data point, descriptors were calculated for each monomer and solvent and concatenated together; the descriptors used for monomers/solvents were not necessarily the same. All classical ML models for homopolymers were near-identical to their copolymer counterparts, with copolymer models adding information about stoichiometry (as an array of the comonomer ratios) and sequencing (as an integer representing random/block/alternating – see Fig. S2 in the ESI†).

The seven descriptor groups used for our classical ML models are referred to by their Python package or by the name of their descriptor class. The molecular descriptors are (1) RDKit, (2) Mordred, and (3) RDKit with Mordred, while the fingerprint descriptors include (4) Morgan fingerprint, (5) RDKit fingerprint, (6) RDKit descriptors with Morgan

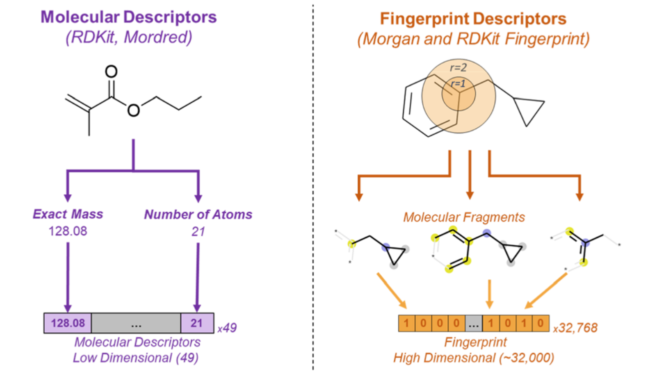

Fig. 1 Representative examples of the molecular and fingerprint descriptors used in this work.

Table 1 An overview of the two polymer solubility databases used for this work

Database name	Data pairs	Good solvents	Bad solvents	Unique polymers	Unique solvents	Best accuracy[a]
Homopolymer	1818	1236	582	431	175	85%
Copolymer	270	200	70	118	43	96%

[a] Percent of correct predictions for each database's best model on a withheld test set.

fingerprint, and (7) RDKit descriptors with RDKit fingerprint. The RDKit descriptors included 25 descriptors for each monomer and 24 descriptors per solvent, with a broad array of molecular descriptors available in the RDKit Python package (*e.g.*, topological shape indices, hydrogen bond acceptors/donors, *etc.*). The pairwise correlations of the RDKit descriptors are shown in Fig. S1 of the ESI,† which also includes the full list of RDKit descriptors. The Mordred descriptors used bear many similarities to the RDKit descriptors in that both describe similar features, but the Mordred descriptors used were much higher dimensions – 885 and 726 descriptors were used to represent each monomer and solvent, respectively. Although we cannot say this conclusively, this does not appear to lead to overfitting based on the 5-fold cross-validation scores (see Results and discussion for additional details) and similarly accurate descriptors with much lower dimensionality (*e.g.*, RDKit). The Morgan and RDKit fingerprint descriptors were used as implemented in the RDKit Python package, with a bit length of 32 768 and a radius of 3. This bit length may appear arbitrary at first glance, but it was chosen after examining multiple other candidate bit lengths which had a smaller bit length and did not cover as much chemical space. In this work, we prioritized chemical space coverage over minimal bit length; this approach appears valid based on our classical model results. All concatenations of different descriptor groups were done linearly (no weighting) without any removal of descriptors except in the combination of RDKit and Mordred, where all descriptors were manually checked to ensure there was no overlap or ‘double-counting’ of the same features in the model. All descriptors were split into train/test sets using the Scikit-Learn python package with a fixed random seed (seed of ‘0’) for reproducibility.[42]

For our graph-based models, descriptor selection was primarily informed by our previous work utilizing a similar architecture to predict small molecule solubility and cetane number.[43–45] The atom, bond, and global features used were comparable to previous work, but the input database and chemical space covered differed significantly.

ML model details

Classical ML models were trained using the Scikit-Learn (version 1.1.2) software package in the Python programming language.[42] Each previously mentioned descriptor group was evaluated over 4 different architectures (RF, DT, naive Bayes, and AdaBoost), which were used with default values as implemented in Scikit-Learn. These models can be subdivided into tree models, ensemble models, and probabilistic models. Tree models make decisions based on branching logic, ensemble models look for an average or consensus among multiple smaller models, and probabilistic models examine probability distributions from input data. Out of the 4 architectures used in the classical ML models, DT can be classified as a tree model, AdaBoost/RF can be classified as (tree-based) ensemble models, and naive Bayes as a probabilistic model. RF and DT were chosen for their interpretability and success, respectively, on small yet diverse datasets. AdaBoost and naive Bayes were selected to probe alternative tree-based models (AdaBoost) and classification schemes (naive Bayes). Some preliminary hyperparameter optimization was performed, but this was not found to impact model performance significantly. A 75/25 train/test split was used with a fixed random seed to ensure reproducible results to evaluate out-of-sample performance for classical ML models. We also performed 5-fold cross-validation on each training set to evaluate model performance across different data subsets; cross-validation metrics presented here are averaged across all five folds. For additional metrics on the classical ML models trained, including confusion matrices and unaveraged cross-validation scores, see Tables S6–S9 in the ESI.†

Graph neural network (GNN) models were trained on a different scheme, with an 80/10/10 train/validation/test split and 5-fold cross-validation (70/20/10); previously published reports inspired the GNN architecture used.[43–45] In our GNN architecture (see Fig. 4), atom/bond/global features are generated for each molecule from their graph representation and then embedded as a 128-dimensional vector (embeddings). These embeddings then undergo a series of message-passing operations to yield a final readout vector. The readout vectors for each monomer–solvent pair are concatenated together and further transformed to produce a soluble/insoluble label. It should be noted that the previously discussed classical model descriptors do not apply to the GNN models presented, as the model input differs too significantly. The GNN models used were constructed and trained in Python 3.8.13 using the following packages: TensorFlow 2.9.1,[46] Keras 2.9.0,[47] RDKit 2022.3.5,[48] and Neural Fingerprint (NFP) 0.3.0.[49] Model metrics for all models were calculated either manually or *via* Scikit-Learn functions.[42] The GNN used categorical cross-entropy as the loss function, and used the Adam optimizer with a learning rate of 1.0×10^{-4}, batch size of 1024, and 1000 epochs.

Experimental solubility measurements

To verify model performance, we evaluated single homopolymer solubility in multiple solvents at 23 °C. Specifically, we examined the solubility of 5 homopolymers (poly(lactic acid) (PLA), polypropylene (PP), poly(methyl methacrylate) (PMMA), polystyrene (PS), poly(acrylic acid) (PAA)) in 4 selected solvents (cyclohexane, dichloromethane (DCM), tetrahydrofuran (THF), toluene). Specific homopolymer and solvent combinations were chosen based on their prevalence in current waste streams and commercial availability while ensuring a diverse spread of model predictions (*i.e.*, there is a similar prevalence of soluble and insoluble predictions). No copolymer experiments were performed due to the additional complexity imposed. PP datapoints were included in our experiments as an intentional challenge to the model's predictive ability and were not part of our model accuracy calculations by design. As the model cannot predict solubility for highly isotactic polymers, it should not be able to effectively predict the solubility of isotactic PP.

In each homopolymer experiment, approximately 250 mg of each polymer was stirred in 10 mL of solvent for two hours; following this, the sample was filtered in a 2.5 μm pore size filter, and the solvent was removed under vacuum. The filtrate and filtered polymer were weighed after drying in a vacuum oven. The mass of filtrate recovered relative to the initial dry polymer mass determined experimental solubility. If more than 10% of the pre-dissolution mass was recovered as filtrate, the data point was labeled soluble (otherwise, insoluble). Any data points with swelling were discarded, and visual observation of solution clarity and homogeneity was used to support any soluble/insoluble labeling. It should be noted that some initial experiments were performed with a different polymer mass (500 mg instead of 250 mg) but at the same concentration (25.0 mg polymer per 1 mL solvent). All experimental results can be found in Table S11 of the ESI.†

SHAP analysis

All SHAP value analyses were performed using the TreeSHAP algorithm, implemented by default in the SHAP Python package.[50–52] All SHAP values shown were calculated on the training set of the homopolymer RF model with RDKit descriptors; test set SHAP values were also calculated for the same model and found to be near-identical to the training set SHAP values in both magnitude and ranking. Due to previously reported issues with the SHAP package on RF classifiers, we manually disabled the additivity check for our SHAP analysis by modifying the discrepancy threshold.

Additive removal

To apply our models to common additives in commercial plastics, we first identified polymer–additive combinations from previous literature, then separately examined the polymer and additive's predicted and literature solubility.[10] Literature additive solubility was gathered from PubChem and manually collected into a database, while literature polymer solubility was determined from our previously described polymer solubility database (see Polymer solubility database above). The preliminary additive solubility database (see Table 2) consisted of two datasets (Dataset A and Dataset B) with 33 and 13 datapoints, respectively.[10] Specifically, from an initial pre-database of approximately 50 additives, we obtained 33 datapoints with solubility data available for the additive and an example polymer and solvent for that additive (Dataset A). Of these 33 datapoints, in 13 cases (Dataset B), there was a difference in solubility between the polymer and additive, allowing for solvent-based additive removal. Lastly, from the 13 cases in Dataset B, we selected 3 example systems where a solvent can drive additive removal from a polymer, as described in Fig. 7.

To predict polymer and additive (small molecule) solubility, our best homopolymer RF model was combined with a newly developed small molecule model, which used the same RF architecture and descriptors as the homopolymer model but was trained on a different database[44] and prediction task (Gibbs free energy of solvation). Additional details on the training and evaluation of this model are located in the ESI† ("Small Molecule Random Forest Performance"). In contrast to the polymer solubility model, the additive solubility model was designed to exclusively predict small molecule solubility, reflected in its differing database and prediction task. All additive solubility predictions had negative $\Delta G_{\text{solvation}}$ and so were assigned a binary solubility label of soluble. This assignment matched the literature additive solubility for 6/9 additives in Dataset A and all additives in Dataset B, with the three exceptions having significant hydrogen bonding (azodicarbonamide and melamine) or high halogen content (decabromodiphenyl ether). The polymer model was used exclusively for polymer solubility prediction, and the additive model was used solely for additive solubility prediction.

Results and discussion

Homopolymer models

As an initial prediction target, we chose to examine the solubility of polymers with one repeat unit (homopolymers). Homopolymers were selected due to their relative abundance and simplicity compared to copolymers, which are much more complex. This work evaluated four architectures for initial predictions of homopolymer solubility, termed 'classical ML models': decision tree (DT), AdaBoost, random forest (RF), and naive Bayes. These classical ML models used molecular and fingerprint descriptors to indirectly represent each data point, with 7 possible descriptor groups for each of the 4 architectures (see Methods) – yielding 28 distinct models (represented by individual bars in Fig. 2). First, considering DT (Fig. 2a), we see similar performance between molecular and fingerprint descriptors, with accuracy ranging from 74 to 79% over all descriptors. The highest accuracy (79%) is by a molecular descriptor (Mordred), which is only 2% less than the most accurate fingerprint descriptor (RDKit FP, 77%). We next examine AdaBoost (Fig. 2b), which, in our case, consists of multiple small decision trees (stubs); given that additional trees can potentially capture different chemical information, one might expect improved performance compared to a single DT. This effect is not particularly strong for AdaBoost as it results in only a ∼2% gain in max accuracy (81%) compared to the DT best accuracy (79%), possibly due to the limited depth of the decision tree stubs our AdaBoost models use. We see more apparent performance gains by comparing the RF models (Fig. 2c) to the DT models, where the best RF model achieves a ∼6% accuracy gain *versus* the best DT model (79%) and a ∼4%

Table 2 Summary of model predictions for additive datapoints with literature solubilities available[10]

Dataset	Description	Datapoints	Correct predictions (polymer)	Correct predictions (additive)
A	Complete set of additive data	33	33	24
B	Data with differing polymer/additive solubility	13	13	13

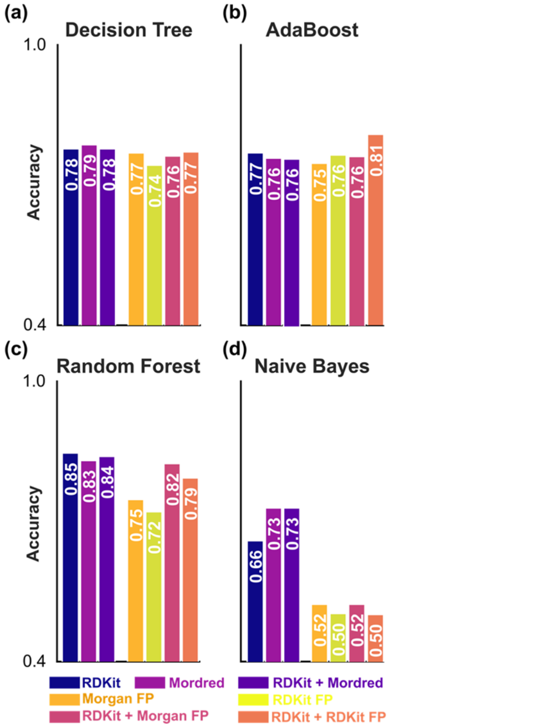

Fig. 2 Homopolymer model test set accuracies for each classical descriptor/architecture combination, including (a) decision tree, (b) AdaBoost, (c) random forest, and (d) naive Bayes. Molecular descriptors are on the left (purple) while fingerprint descriptors are on the right (orange).

gain *vs.* the best AdaBoost model (81%). Explanations for RF's success compared to AdaBoost include increased depth of trees (1 for AdaBoost *vs.* unlimited for RF), increased number of estimators (50 for AdaBoost *vs.* 100 for RF), or the addition of random subsampling in RF (not present in AdaBoost). Compared to DT, both AdaBoost and RF show an increased range in performance (5/6/13%, respectively) but a higher maximum accuracy (79/81/85%). In great contrast to these tree-based models, the probabilistic naive Bayes models (Fig. 2d) have a massive accuracy range (23%) with low maximum accuracy (73%). While these issues may be related to dataset distribution or data processing, it is difficult to conclude about the performance given the very low performance for fingerprint descriptors, which is almost on par with a random solubility assignment of 50%. In summary, from Fig. 2 we see that the RF architecture performs the best with the highest accuracy (85%) out of all models; AdaBoost (81%) and DT (79%) lag somewhat while naive Bayes (73%) performs poorly in most cases.

In addition to architecture-level comparisons, Fig. 2 also examines the test set accuracy of each descriptor-derived model. We see that the molecular descriptors (Mordred, RDKit) and their combinations significantly outperform fingerprint-based descriptors on average, with the RDKit descriptors performing the best using a RF architecture (85% accuracy). The differences between each category's top performers range from <5% to over 20%, demonstrating significant variance in descriptor performance across different architectures. While in some cases (AdaBoost, DT), the fingerprint descriptors outperform the molecular descriptors, the peak fingerprint performance over all models is still 3% lower than the peak molecular performance. Furthermore, for all architectures except naive Bayes, adding molecular descriptors to fingerprint descriptors (the two rightmost bars) yields higher or near-equivalent accuracy than the uncombined fingerprint descriptors. This can be rationalized by considering that the top 4 performing descriptors (RDKit, RDKit + Mordred, Morgan FP + RDKit, RDKit FP + RDKit) contain both property-based and structural information, in contrast to the uncombined fingerprint descriptors, which only have structural information. For instance, uncombined fingerprint descriptors can only implicitly encode molecular properties such as the number of hydrogen bond donors or aromatic rings, which can be highly relevant to solubility. Additionally, the top-performing fingerprint descriptors represent their combination with molecular descriptors – by omitting these combinations, the highest fingerprint accuracy is only 77% (Fig. 2a). Comparatively, the most accurate molecular descriptor models can achieve 85% accuracy, using solely RDKit descriptors with a RF architecture.

We can also analyze model performance relative to the dimensionality of each descriptor. Descriptor dimensionality is relevant to model prediction as descriptors with high dimensionality can potentially fail to generalize (overfit), weakening a key advantage of machine learning over alternative methods. We find molecular descriptors are relatively low dimensional (49 for RDKit, 1611 for Mordred), whereas fingerprint descriptors are relatively high dimensional (∼32 000). Here, the descriptor dimensionality is determined by the number of entries needed to describe an individual datapoint to the model, which is generally the number of monomer descriptors plus the number of solvent descriptors (*e.g.* there are 25 + 24 = 49 descriptors for the RDKit models and 885 + 726 = 1611 descriptors for the Mordred models). For the molecular descriptors (Fig. 2a–d), we see a <2% accuracy difference between the RDKit and Mordred descriptors for all architectures except naive Bayes, despite a 30× increase in dimensionality. As there is a significant dimensionality increase with minimal change in accuracy, we argue that the dimensionality of the Mordred descriptors is appropriate. Moving to the fingerprint descriptors (Fig. 2a–d), we see a more significant variance in accuracy (2–10%) between descriptors despite relatively constant dimensionality (adding RDKit and Mordred to fingerprints increases dimensionality by <5%).

We performed a cross-validation analysis on the best-performing RF architecture to further analyze model generalizability. Specifically, in Fig. 3, we compare each RF model's test set accuracy to its k-fold cross-validation score, averaged over five folds of the model training set ($k = 5$). This analysis shows that the average cross-validation scores are high (>75%) and within 5% of the test set accuracy (Fig. 2) for all descriptors. Furthermore, for these RF models, we find that molecular descriptors outperform the fingerprint descriptors in every case, with the lowest dimensional molecular descriptor model (RDKit) yielding a high cross-validation score (82%). The remaining molecular models (RF with Mordred and RF with

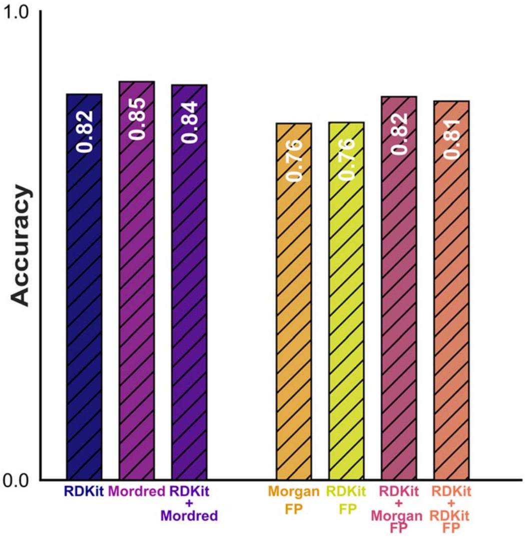

Fig. 3 Comparison of 5-fold cross-validation score for all random forest models of homopolymer solubility. For each descriptor, striped bars represent the average model accuracy over 5 folds of cross-validation.

RDKit + Mordred) have marginally higher (2–3%) cross-validation scores but have much higher dimensionality, as previously discussed (49 for RDKit *vs.* 1611 for Mordred). Compared to the molecular descriptor models, the two uncombined fingerprint models (Morgan FP, RDKit FP) have much lower cross-validation scores (76%/76%). Adding RDKit descriptors to the Morgan FP and RDKit FP models improves their cross-validation scores by at least 5%. Additionally, the difference between their test set accuracy and their cross-validation score decreases – implying that adding molecular descriptors to fingerprint models makes these models more accurate and representative of the entire dataset, agreeing with our comments on Fig. 2 above. This analysis concludes that a RF model with RDKit descriptors is precise, highly accurate, and potentially generalizable in predicting homopolymer solubility. Adding molecular descriptors to fingerprint-based models can improve model accuracy and generalizability.

To ensure that our homopolymer models applied to real-world plastics, we performed in-house experiments to measure the solubility of commercial plastics. These experiments examined the dissolution of single homopolymers in select solvents, with the homopolymers and solvents chosen based on their ubiquity. Our results are summarized in Table S11.† Our best classical homopolymer ML model (RF with RDKit descriptors) achieved 79% experimental accuracy, demonstrating the predictive power of our RF model when combined with the easily calculated and low-dimensional RDKit descriptors chosen.

In addition to the classical homopolymer ML models, a more sophisticated ML model (a graph neural network, GNN) was also used to predict homopolymer solubility. Previous work from our group (predicting bond dissociation enthalpy, small molecule solubility, and cetane number) inspired the GNN architecture shown in Fig. 4.[43–45] Unlike their classical counterparts, graph neural networks (GNNs) are a class of neural networks that use chemical bonds and atom information directly as model input rather than indirectly through descriptors. In addition to atom and bond information, other global descriptors can be provided to add chemical context. In this case, we use the number of hydrogen bond acceptors/donors, the Labute accessible surface area (ASA), and the topological polarizable surface area (TPSA) as global descriptors for solubility. These descriptors were chosen for their relevance to solubility. For example, hydrogen bonding between a solute and solvent can provide a robust enthalpic contribution to the free energy of dissolution, compounded by the fact that these intermolecular interactions are significant for long polymer chains. The Labute ASA approximates the available surface area for solvent interaction, while the TPSA can approximate membrane permeability.[53] The TPSA, in particular, has been used in previous models of polymer solubility.[54] By using these molecular-level global descriptors alongside the atom-and-bond level molecular graph, our GNN is believed to capture both local and global solubility information.

We find a 5-fold cross-validation accuracy of 81% for the GNN, compared to 82% for the best classical model (RF with RDKit descriptors). The similar performance between the GNN and our best classical model is somewhat surprising, especially given that the RF model with RDKit descriptors has far fewer descriptors and model parameters than the GNN. One possible explanation is that the input dataset is not yet large enough for the GNN to learn polymer solubility deeply and adjust parameters. Further comparisons with the classical case are challenging, as our GNN only has one architecture which cannot be trivially separated into individual components. Nevertheless, we developed this deep model to predict polymer solubility as it possesses multiple advantages over its classical counterparts, and a similar architecture has succeeded in predicting small molecule solubility.[44] The first GNN advantage is flexibility: most classical ML models cannot be trivially modified without undoing previous model refinement and development, while most GNNs can be modified to suit the task or dataset.

Additionally, it is expected that deep model performance scales quite well with database size – thus, we anticipate that with more input data points, the GNN model will outperform its classical counterparts, which are generally superior for smaller datasets. Lastly, deep ML models simply have more (hyper) parameters to tune, allowing for them to potentially better approximate high dimensional problems. With this in mind, we present the developed GNN as a proof-of-concept that can undergo significant fine-tuning to (potentially) improve accuracy over current classical ML models.

Homopolymer model analysis

To explain our models and their predictions, a SHAP value analysis was performed on the most accurate homopolymer model.[50,51] SHAP values estimate the impact of each descriptor on model performance, providing valuable insights into explaining model output as well as into the most impactful

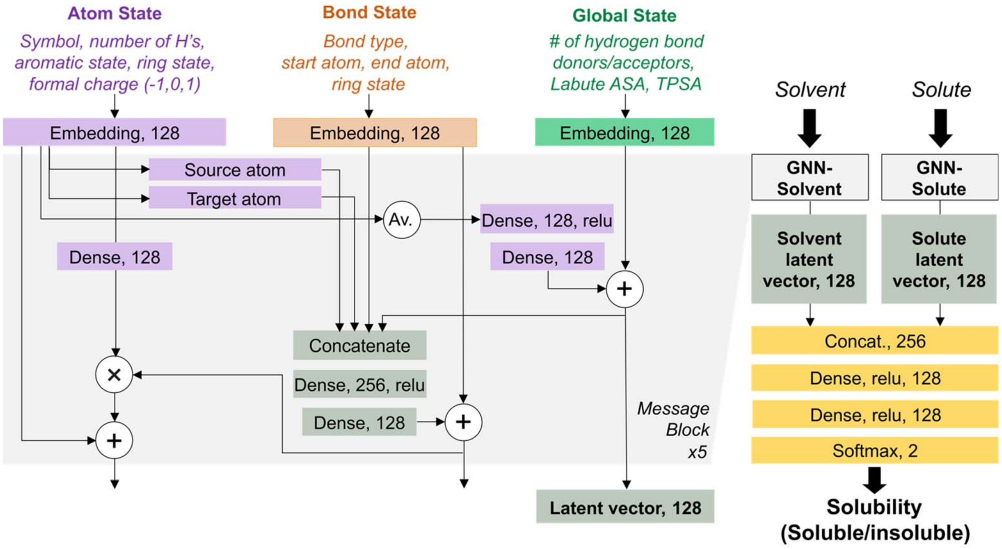

Fig. 4 Overview of the GNN architecture used for homopolymer solubility.[43–45]

descriptors (those with the largest SHAP values). SHAP analysis can be applied to individual data points and entire datasets, making it a flexible model explanation and analysis tool. We applied SHAP analysis using predictions from the best homopolymer model (RF with RDKit descriptors) to understand our homopolymer model performance better.

To explain experimental performance, we first analyzed the predicted solubility of polystyrene in toluene and poly(lauryl methacrylate) in ethyl acetate (Fig. 5). Our model prediction is accurate in both cases, though the most impactful features differ significantly in magnitude and type. We first consider the solubility of polystyrene in toluene. As expected for a polymer with a pendant aromatic ring, it is unsurprising that polystyrene dissolves in toluene. In terms of model predictions, solvent connectivity/shape indices (ChiNv and kappaN) and monomer TPSA significantly increase the probability of a soluble prediction (P (soluble)), leading to an approximately 99% chance that the model will predict the combination to be soluble (represented by P (soluble) in Fig. 5). In contrast, the fraction of sp^3 carbons in the monomer (0.00), number of heteroatoms in the solvent (0), and Chi4v of the monomer (0.59) decrease P (soluble), though this has a much lower cumulative impact than the solvent connectivity/shape indices – leading to the high net P (soluble) of 0.99. Since styrene and toluene have aromatic rings that can favorably interact, this prediction result is reasonable and agrees with our experimental results and our input database of polymer solubility.

We next examine the solubility of poly(lauryl methacrylate) in ethyl acetate. As opposed to the polystyrene/toluene combination, monomer features followed by solvent features contribute significantly to a low P (soluble) of 0.35. This leads to an insoluble prediction that agrees with our input database.

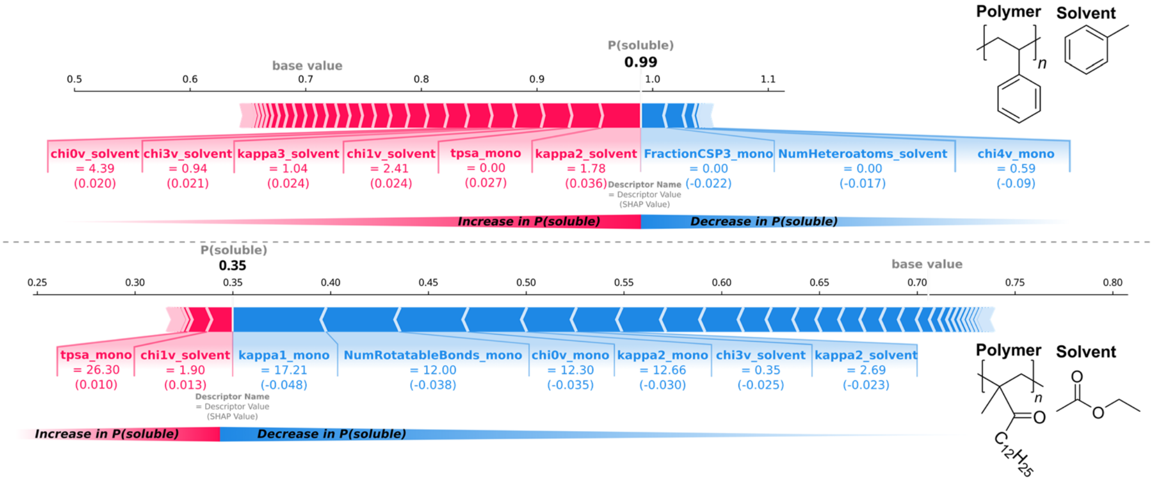

Fig. 5 SHAP values for polystyrene/toluene (top) and poly(lauryl methacrylate)/ethyl acetate (bottom). A positive SHAP value (red) increases the likelihood of soluble prediction, while a negative SHAP value increases the likelihood of insoluble prediction.

Monomer connectivity/shape indices and many rotatable bonds are the most impactful features, and they strongly decrease P (soluble), while the monomer TPSA (26.30) and solvent chi1v (1.90) moderately increase the probability of a soluble prediction. This can be rationalized by examining the structure of the monomer and solvent. Lauryl methacrylate has a long twelve-carbon chain, which significantly impacts the value of its connectivity/shape indices and the number of rotatable bonds, supporting the importance of these features. Ethyl acetate, on the other hand, is a relatively small molecule with only two C2 chains, represented by the solvent's chi3v (0.35) and kappa2 (2.69). As one might expect, monomer and solvent shape play an important role in solubility for these two cases – but whether these trends hold for the entire dataset remains unclear.

To answer this question, we also performed an aggregate SHAP analysis (Fig. 6). We find that the trends in the polystyrene/toluene case generally hold for the entire dataset, as solvent connectivity/shape indices and number of heteroatoms have the largest SHAP values followed by monomer TPSA, monomer fraction of sp^3 carbons, and solvent Lipinski hydrogen bond donors. Since solvents are much smaller and generally diffuse faster than long polymer chains, solvent shape, and connectivity descriptors play an important role in model prediction. Monomer shape and connectivity descriptors are also impactful as they make up the majority of the 11th to 20th largest SHAP values (not shown), but they have smaller average SHAP values than solvent shape/connectivity descriptors, which are within the top 10 largest SHAP values and make up all of the top 5 (Fig. 6a).

In addition to the averaged analysis in Fig. 6a, we consider the distribution of SHAP values as feature values change (Fig. 6b). From this, we see distinct relationships between high/low feature values and positive/negative SHAP values, with vertical line widths representing population density. For solvent chi1v and chi0v, we observe that low feature values typically decrease the probability of a soluble prediction, whereas the opposite trend is present for solvent kappa2. While solvent kappa1 and kappa3 show similar trends to solvent chi1v/chi0v, the number of heteroatoms in the solvent shows two distinct clusters. This feature often minimally decreases P (soluble) in solvents with a low number of heteroatoms, but the overall P (soluble) increases in solvents with a larger number of heteroatoms.

In comparison, a significant feature value for monomer TPSA decreases the probability of a soluble prediction significantly, whereas a low value has an opposite and lesser impact, possibly due to a mismatch between monomer and solvent TPSA distributions in our dataset. The fraction of sp^3 carbons in the monomer has a much less pronounced impact, with the bulk of data points having minimal SHAP values. Still, there are cases for which this descriptor becomes more important, as seen by the tailing for both positive and negative SHAP values. Lastly, the number of Lipinski hydrogen bond donors per solvent minimally increases P (soluble) at low values but moderately decreases P (soluble) at high values – this is theorized to be due to a mismatch in polarity between monomer and solvent as in the TPSA case.

From the SHAP analysis above, we can consider the most impactful model features and general trends in feature contributions. We find that, on average, the solvent connectivity, shape, and number of heteroatoms contribute the most to polymer solubility, followed by the monomer TPSA and fraction of sp^3 carbons. Surprisingly, less impactful is the solvent's number of hydrogen bond donors (Lipinski), potentially because mismatches between solvent branching/polarity and monomer branching/polarity could lead to limited solvent diffusion – which severely limits the number of hydrogen bonds that can form. Although SHAP values measure feature contributions to model predictions rather than to experimental solubility, given the high accuracy of our model on a diverse set of monomers and solvents (431/175 unique, respectively), we have confidence that we can use our SHAP analysis to propose design strategies for polymer solubility. Specifically, we recommend that a good solvent for a polymer should have a similar shape and degree of branching to the polymer's monomer (Fig. 5), and ideally, the polymer's monomer should have a low TPSA value (Fig. 6). As previously stated, solvent shape and degree of branching appear significantly more

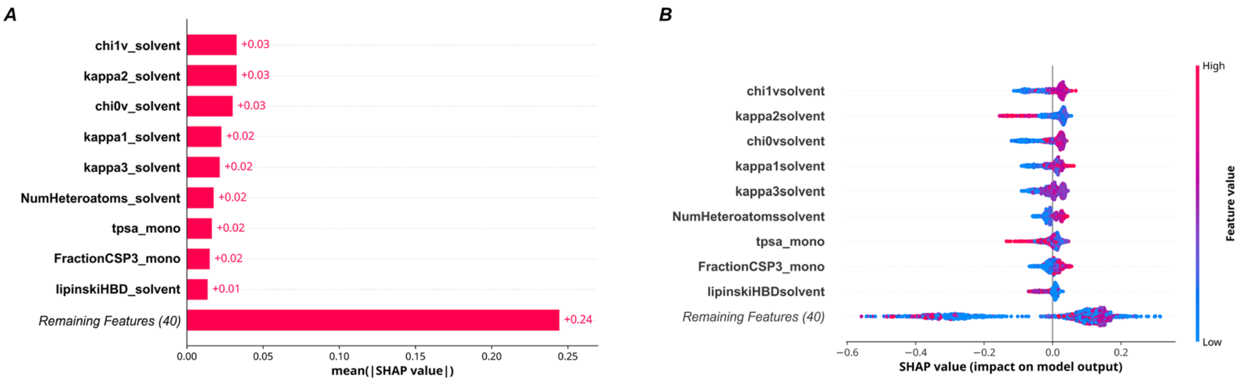

Fig. 6 (A) Average SHAP values for the top 10 descriptors in the RF with RDKit descriptors model for homopolymer solubility. (B) Beeswarm plots for the same descriptors/models as in A. Vertical linewidth corresponds to density of points, *x* axis position corresponds to SHAP value, and color correspond to feature value.

important than the number of hydrogen bond donors/acceptors. Our recommendations for choosing a poor solvent follow a similar rationale to the good solvent case, and we recommend that a poor solvent should have a different shape and degree of branching compared to the polymer's monomer. Our findings agree with the familiar adage of 'like dissolves like' when considering molecular shape and connectivity, affirming that the recommendations from our SHAP analysis make chemical sense.

Application: ML-predicted solvents for additive removal

To demonstrate the utility of our polymer solubility models, we used our best homopolymer model to identify potential solvents for polymer additive removal *via* dissolution (Fig. 7). Specifically, we constructed a database of polymer/additive pairings from previous literature,[10] excluded datapoints without literature solubility available, and then predicted the solubility of each polymer/additive for two preliminary datasets (Table 2 in Methods). We used our best homopolymer model (RF with RDKit) to predict polymer solubility while using a new additive-specific RF model to predict additive solubility (see Methods). From our solubility predictions on the preliminary datasets, we selected solvents for additive removal from 3 polymer/additive pairings; these 3 cases are highlighted in Fig. 7. The polymers selected are highly prevalent as commercial plastics and commonly contain additives, with the additives used ranging from non-toxic food additives (stearic acid[55]) to potentially carcinogenic or genotoxic azo dyes (Sudan I[56]) (Fig. 7).

We first examine our polymer-additive-solvent system for polystyrene, one of the most well-studied polymers, which was chosen here for its relative simplicity. In this case, the polystyrene additive (stearic acid) is a fatty acid added to polystyrene as a lubricant.[10] This lubricant reduces internal or external friction for the polymer, preventing polymer films from sticking or decreasing thermal damage under high shear stress.[10] While stearic acid is non-toxic,[55] any additive presence limits future recycling or upcycling applications, motivating its removal. To remove stearic acid, we utilize solid–liquid extraction by dissolving the additive but not polystyrene in diethyl ether, separating this solution from the polymer, and precipitating stearic acid from the solution by either solvent evaporation, cooling, or by addition of water.[10] Towards this, our small molecule and polymer solubility models agree with literature data, and we thus believe that our proposed solvent system is reasonable.

In addition to polystyrene, we predicted two selective solvent systems for polyethylene, which, like polystyrene, is a cornerstone of modern materials. However, the prevalence of polyethylene and the breadth of its applications make recycling and proper valorization difficult yet important. To limit the scope of these challenges, in this example, we limit ourselves to the study of minimally-branched polyethylene (HDPE), modeled as its ethylene monomer. While the material properties of branched PE have spawned rich applications in many industries, the broader aim of our polymer solubility models is to capture chemical effects first rather than the effects of the data-scarce material, such as polymer chain branching (which are not yet sufficiently captured in our dataset). Despite this challenge, we report two examples where polyethylene is separated from two potentially hazardous additives, Sudan I and 2,4-dihydroxybenzophenone.[56–59]

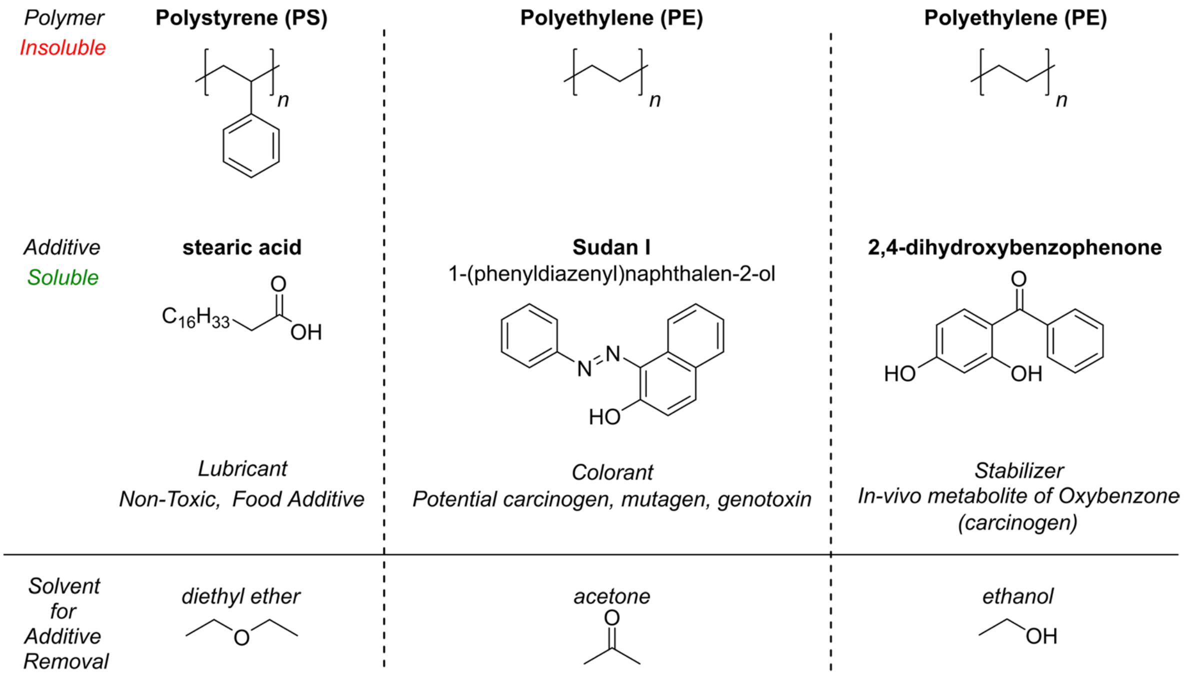

Fig. 7 Selected systems for solvent-based additive removal from common polymers. For each polymer and additive combination, our machine learning models identified solvents which would selectively dissolve the additive but not the polymer. For all systems, the polymer is insoluble in the chosen solvent while the additive dissolves in the chosen solvent.

The first system removes a colorant (Sudan I) from polyethylene by adding acetone and precipitating the additive using the same method as the polystyrene example above. Colorants are additives that change the visible color of polymers and potentially increase heat/light stability.[10] While colorants such as Sudan I may be well-suited for their initial application, colorant removal is vital to ensuring optical clarity and consistency in recycled polymers, and Sudan I itself is a potential carcinogen, mutagen, and genotoxin.[56]

The second system presented uses ethanol to dissolve 2,4-dihydroxybenzophenone, a stabilizer. Stabilizers generally increase polymer resistance to light and heat in a more targeted and effective manner than colorants, potentially improving or maintaining mechanical properties.[10] While this is generally beneficial, stabilizers such as 2,4-dihydroxybenzophenone may pose health or environmental risks. A related compound used in sunscreens, oxybenzone, has been banned in Hawaii for its potential danger to coral, and oxybenzone metabolizes to 2,4-dihydroxybenzophenone *in vitro*.[57–59] While this is not conclusive evidence of risks in using 2,4-dihydroxybenzophenone, it is enough to merit an investigation of its removal. We, therefore, propose the removal of 2,4-dihydroxybenzophenone from polyethylene by the addition of ethanol followed by precipitation, as previously discussed.

While the above examples may appear trivial at first glance, given the deep well of knowledge regarding PE and PS, there is no fundamental limitation to applying our methodology to arbitrary polymers with arbitrary solvents other than data scarcity. Using a data-driven approach rather than intuition or prior knowledge, one could identify multiple candidate solvents to remove banned or hazardous additives from multiple polymers within hours rather than weeks or months of testing. Furthermore, one can use our approach in predictive modeling for materials design by identifying additive removal pathways at the design stage rather than after a plastic has been produced. Lastly, while in this work, we focus on solid–liquid extraction, where only the additive dissolves, we can also use our models for dissolution-precipitation, where both the polymer and additive dissolve. Following dissolution, adding a nonsolvent or solvent evaporation change causes polymer precipitation – leading to the effective removal of the additive.[10] This method would potentially be more efficient for densely packed or high molecular weight polymers but was not further examined due to limited data.

Copolymer models

In contrast to homopolymer solubility, ML has largely remained unapplied to copolymer solubility in the literature. While multiple reports predict copolymer thermal, electronic, and morphological properties, there do not appear to be any reports that predict copolymer solubility separately from homopolymers for a diverse set of copolymers using ML.[30,34,37,60–63] In this work, we report the first prediction of copolymer solubility using machine learning and have successfully predicted solubility for copolymers with two repeat units at over 90% accuracy on a test set. To achieve this, we adapted the RF with RDKit descriptors model for homopolymer solubility to copolymeric systems (Fig. 8), as the RF architecture achieved high accuracy with few descriptors. Rather than attempt to derive a more fundamental relationship between copolymer monomers and solubility, we chose to calculate monomer descriptors for each comonomer and concatenate these descriptors together with the solvent descriptors – analogous to the homopolymer models, but with two monomers. Different from the homopolymer models, however, was the incorporation of copolymer ratio and sequencing information (*e.g.*, random block) into the model input; this information was also concatenated with the monomer and solvent descriptors to form the final model input.

From the cross-validation results in Fig. 8, all the descriptors examined achieve a cross-validation score of over 75%. We also see similar trends in descriptor performance compared to the homopolymer RF models. This is somewhat surprising, as even though the model architecture (RF) and prediction target (polymer solubility) have not changed, the database and model input have. First, considering molecular descriptors, we find that RDKit again leads to the best balance of model accuracy and descriptor dimensionality, as its cross-validation score is only 1% below the much higher-dimensional Mordred and RDKit + Mordred models. As in the homopolymer case, the uncombined fingerprint models (Morgan FP, RDKit FP) achieve the lowest overall performance at 81% and 77% cross-validation scores. Furthermore, we see the same trend in the combined fingerprint models as before in the homopolymer models – adding RDKit descriptors to fingerprint models improves their cross-validation score by at least 5%. This further supports the claim that our RF models and descriptors are robust and generalizable, as we see near identical trends in descriptor performance despite significantly different database and model inputs. Although the improved accuracy of the copolymer models compared to the homopolymer models may appear contradictory, the smaller copolymer database is less chemically diverse compared to the larger homopolymer database.

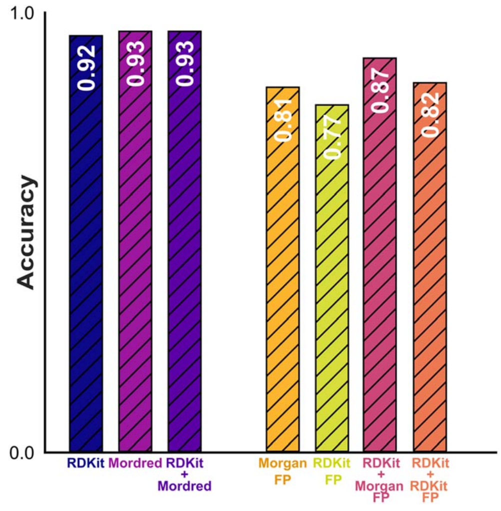

Fig. 8 Averaged 5-fold cross-validation scores for the RF copolymer models.

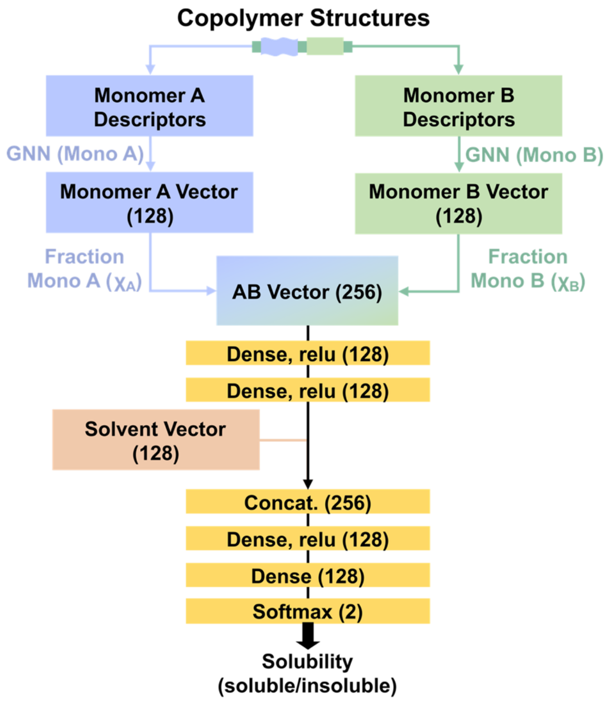

Fig. 9 The modified copolymer architecture used for the copolymer GNN model. Note that the message passing structure and some solvent vector operations are omitted for clarity. See Fig. 4 for additional model details, such as the conversion of monomer/solvent descriptors into a 128-dimensional vector.

This makes for an easier prediction task which leads to higher model accuracy, explaining why our copolymer models perform well despite the additional structural complexity of copolymers.

To further expand the scope of copolymer architectures investigated, we also developed a new GNN architecture to predict copolymer solubility. Compared to our homopolymer GNN model, we use three neural networks (one for each component) instead of two, and we also use a weighted average to combine the two comonomer neural networks, taking into account the copolymer ratio (Fig. 9). As far as the authors know, this approach to predicting copolymer solubility has not been reported in the current literature. While the reported 5-fold cross-validation score of the copolymer GNN model (86%) is lower than the copolymer RF model with RDKit descriptors (92%), the presented GNN architecture has not been subject to years of refinement as the most common classical ML models have. Furthermore, similar to our homopolymer GNN, model performance is expected to improve with database size. Given this, GNN models of homopolymer and copolymer solubility may have the potential to surpass their classical counterparts, given sufficient honing.

Conclusions

In this work, we developed multiple highly accurate machine-learning models to predict homopolymer and copolymer solubility. We created two novel homopolymer and copolymer solubility databases from previous literature with 1818 and 270 datapoints, respectively. We examine a wide range of architectures (AdaBoost, decision tree, naive Bayes, random forest, graph neural networks) and descriptors and achieve average 5-fold cross-validation accuracies of 82% (homopolymer random forest) and 92% (copolymer random forest). We experimentally validate our homopolymer model on commercial plastics and achieve 79% accuracy on these experimental predictions. We then characterize this homopolymer model's performance using SHAP analysis, which revealed that solvent shape and degree of branching play a crucial role in our model predictions. Lastly, we apply our homopolymer model to remove additives from common waste plastics by identifying solvents that remove additives using solid–liquid extraction and selective dissolution. Overall, this work represents a novel contribution toward better understanding and predicting polymer solubility through machine learning, with demonstrable and relevant real-world applications. Future work will expand the polymer solubility databases used, consider polymer morphology and crystallinity in ML predictions, and incorporate polymer and solvent diffusion modelling.

Data availability

Data for this article, including the databases developed, code used to generate ML descriptors, train all ML models, and perform all analysis, are available at **https://doi.org/10.5281/zenodo.14376748** (**https://github.com/cstubb/PolySol**). The homopolymer and copolymer databases used to train all models are available at the same link.

Author contributions

C. D. S. was responsible for data curation, conceptualization, formal analysis, investigation, methodology, software development, validation, and visualization, and wrote the initial draft of the manuscript. Y. K. contributed to conceptualization, investigation, methodology, and visualization; Y. K. also provided the initial code for the reported GNN models. R. P.-S. was involved in formal analysis, investigation, methodology, and validation. E. C. Q. and C. D. S. chose the experimental validation scope and procedure, while E. C. Q refined this procedure, performed the necessary experiments, and verified the experimental results. E. Y.-X. C. and S. K. provided project administration and supervision, conceptualization, validation, and funding acquisition. All authors participated in editing the final manuscript.

Conflicts of interest

The authors report no conflicts of interest.

Acknowledgements

The authors would like to graciously acknowledge support from the Department of Chemistry at Colorado State University. This work was also supported by a National Science Foundation grant (NSF CHE-2304658). The work was supported by the U.S. Department of Energy, Office of Energy Efficiency and

Renewable Energy, Advanced Materials and Manufacturing Technologies Office (AMMTO) and Bioenergy Technologies Office (BETO), performed as part of the BOTTLE™ Consortium and funded under contract no. DE-AC36-08GO28308 with the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy. Computational resources were provided by the NSF Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support program (ACCESS), Grant No. TG-CHE210034.

Notes and references

1 P. Borgquist, A. Körner, L. Piculell, A. Larsson and A. Axelsson, *J. Controlled Release*, 2006, **113**(3), 216–225.

2 J. P. Robinson, E. S. Tarleton, C. R. Millington and A. Nijmeijer, *J. Membr. Sci.*, 2004, **230**(1), 29–37.

3 W. C. Li, H. F. Tse and L. Fok, *Sci. Total Environ.*, 2016, **566–567**, 333–349.

4 S. Sharma and S. Chatterjee, *Environ. Sci. Pollut. Res.*, 2017, **24**(27), 21530–21547.

5 R. Geyer, J. R. Jambeck and K. L. Law, *Sci. Adv.*, 2017, **3**(7), e1700782.

6 C. Jehanno, J. W. Alty, M. Roosen, S. De Meester, A. P. Dove, E. Y.-X. Chen, F. A. Leibfarth and H. Sardon, *Nature*, 2022, **603**(7903), 803–814.

7 J. Payne and M. D. Jones, *ChemSusChem*, 2021, **14**(19), 4041–4070.

8 I. Vollmer, M. J. F. Jenks, M. C. P. Roelands, R. J. White, T. van Harmelen, P. de Wild, G. P. van der Laan, F. Meirer, J. T. F. Keurentjes and B. M. Weckhuysen, *Angew. Chem., Int. Ed.*, 2020, **59**(36), 15402–15423.

9 Y.-B. Zhao, X.-D. Lv and H.-G. Ni, *Chemosphere*, 2018, **209**, 707–720.

10 S. Ügdüler, K. M. Van Geem, M. Roosen, E. I. P. Delbeke and S. De Meester, *Waste Manage.*, 2020, **104**, 148–182.

11 T. W. Walker, N. Frelka, Z. Shen, A. K. Chew, J. Banick, S. Grey, M. S. Kim, J. A. Dumesic, R. C. Van Lehn and G. W. Huber, *Sci. Adv.*, 2020, **6**(47), eaba7599.

12 M. Doi and H. See, *Introduction to Polymer Physics*, Clarendon Press, 1996.

13 B. A. Miller-Chou and J. L. Koenig, *Prog. Polym. Sci.*, 2003, **28**(8), 1223–1270.

14 J. Jimenez and E. Ford, *Polymer*, 2021, **230**, 124079.

15 Y. Tao, B. D. Olsen, V. Ganesan and R. A. Segalman, *Macromolecules*, 2007, **40**(9), 3320–3327.

16 Y. Liu, T. Zhao, W. Ju and S. Shi, *J. Mater.*, 2017, **3**(3), 159–177.

17 R. Ramprasad, R. Batra, G. Pilania, A. Mannodi-Kanakkithodi and C. Kim, *npj Comput. Mater.*, 2017, **3**(1), 1–13.

18 K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev and A. Walsh, *Nature*, 2018, **559**(7715), 547–555.

19 J. E. Saal, A. O. Oliynyk and B. Meredig, *Annu. Rev. Mater. Res.*, 2020, **50**(1), 49–69.

20 L. H. Hall and L. B. Kier, in *Reviews in Computational Chemistry*, ed. K. B. Lipkowitz and D. B. Boyd, John Wiley & Sons, Inc., Hoboken, NJ, USA, 1991, pp 367–422.

21 M. Chi, R. Gargouri, T. Schrader, K. Damak, R. Maâlej and M. Sierka, *Polymers*, 2022, **14**(1), 26.

22 T.-L. Liu, L.-Y. Liu, F. Ding and Y.-Q. Li, *Chin. J. Polym. Sci.*, 2022, **40**(7), 834–842.

23 E. Terrell, *Chem. Eng. Sci.*, 2022, **248**, 117184.

24 S. Venkatram, C. Kim, A. Chandrasekaran and R. Ramprasad, *J. Chem. Inf. Model.*, 2019, **59**(10), 4188–4194.

25 A. Chandrasekaran, C. Kim, S. Venkatram and R. Ramprasad, *Macromolecules*, 2020, **53**(12), 4764–4769.

26 J. Kern, S. Venkatram, M. Banerjee, B. Brettmann and R. Ramprasad, *Phys. Chem. Chem. Phys.*, 2022, **24**(43), 26547–26555.

27 Z. Wang, C. L. C. Chan, T. H. Zhao, R. M. Parker and S. Vignolini, *Adv. Opt. Mater.*, 2021, 2100519.

28 H.-C. Kim, S.-M. Park and W. D. Hinsberg, *Chem. Rev.*, 2010, **110**(1), 146–177.

29 C. Li, Q. Li, Y. V. Kaneti, D. Hou, Y. Yamauchi and Y. Mai, *Chem. Soc. Rev.*, 2020, **49**(14), 4681–4736.

30 K. K. Bejagam, J. Lalonde, C. N. Iverson, B. L. Marrone and G. Pilania, *J. Phys. Chem. B*, 2022, **126**(4), 934–945.

31 A. Boublia, T. Lemaoui, J. AlYammahi, A. S. Darwish, A. Ahmad, M. Alam, F. Banat, Y. Benguerba and I. M. AlNashef, *ACS Sustainable Chem. Eng.*, 2023, **11**(1), 208–227.

32 Z. Jiang, J. Hu, B. L. Marrone, G. Pilania and X. Yu Bill, *Materials*, 2020, **13**(24), 5701.

33 T. Aoyagi, *Comput. Mater. Sci.*, 2022, **207**, 111286.

34 J. A. Pugar, C. Gang, C. Huang, K. W. Haider and N. R. Washburn, *ACS Appl. Mater. Interfaces*, 2022, **14**(14), 16568–16581.

35 Z. Feng, Y. Cheng, A. Khlyustova, A. Wani, T. Franklin, J. D. Varner, A. L. Hook and R. Yang, *Adv. Mater. Technol.*, 2023, **8**(13), 2201533.

36 S. Zhao, T. Cai, L. Zhang, W. Li and J. Lin, *ACS Macro Lett.*, 2021, **10**(5), 598–602.

37 K.-H. Tu, H. Huang, S. Lee, W. Lee, Z. Sun, A. Alexander-Katz and C. A. Ross, *Adv. Mater.*, 2020, **32**(52), 2005713.

38 A. Arora, T.-S. Lin, N. J. Rebello, S. H. M. Av-Ron, H. Mochigase and B. D. Olsen, *ACS Macro Lett.*, 2021, **10**(11), 1339–1345.

39 P. Zhou, J. Yu, K. L. Sánchez-Rivera, G. W. Huber and R. C. V. Lehn, *Green Chem.*, 2023, **25**(11), 4402–4414.

40 P. Zhou, K. L. Sánchez-Rivera, G. W. Huber and R. C. Van Lehn, *ChemSusChem*, 2021, **14**(19), 4307–4316.

41 J. Brandrup, E. H. Immergut and E. A. Grulke, *Polymer Handbook*, Wiley, New York, 4th edn, 2004.

42 F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and É. Duchesnay, *J. Mach. Learn. Res.*, 2011, **12**(85), 2825–2830.

43 P. C. St. John, Y. Guan, Y. Kim, S. Kim and R. S. Paton, *Nat. Commun.*, 2020, **11**(1), 2328.

44 Y. Kim, H. Jung, S. Kumar, R. S. Paton and S. Kim, *Chem. Sci.*, 2024, **15**(3), 923–939.

45 Y. Kim, J. Cho, N. Naser, S. Kumar, K. Jeong, R. L. McCormick, P. C. St. John and S. Kim, *Proc. Combust. Inst.*, 2023, **39**(4), 4969–4978.

46 M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu and X. Zheng, *arXiv*, 2016, perprint, arXiv:1603.04467, DOI: **10.48550/arXiv.1603.04467**.

47 A. Gullì and S. Pal, *Deep learning with Keras: implement neural networks with Keras on Theano and TensorFlow*, Packt Publishing, Birmingham Mumbai, 2017.

48 G. Landrum, P. Tosco, B. Kelley, Ric, Sriniker, Gedeck, R. Vianello, NadineSchneider, E. Kawashima, A. Dalke, D. Cosgrove, N. D., G. Jones, B. Cole, M. Swain, S. Turk, AlexanderSavelyev, A. Vaucher, M. Wójcikowski, I. Take, D. Probst, K. Ujihara, V. F. Scalfani, Godin, Guillaume, A. Pahl, F. Berenger, JLVarjo and strets123, J. P., DoliathGavid, *rdkit/rdkit: 2022_03_4 (Q1 2022) Release*, 2022.

49 P. St. John, L. Ward and S. V. Shree Sowndarya, *NFP (Neural Fingerprint) 0.3.0*, 2019.

50 S. M. Lundberg and S.-I. Lee, in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2017, vol. 30.

51 S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal and S.-I. Lee, *Nat. Mach. Intell.*, 2020, **2**(1), 56–67.

52 L. S. Shapley, in *17. A Value for n-Person Games*, Princeton University Press, 2016, pp. 307–318.

53 P. Ertl, B. Rohde and P. Selzer, *J. Med. Chem.*, 2000, **43**(20), 3714–3717.

54 J. Nistane, L. Chen, Y. Lee, R. Lively and R. Ramprasad, *MRS Communications*, 2022.

55 EFSA Panel on Food Additives and Nutrient Sources added to Food (ANS), A. Mortensen, F. Aguilar, R. Crebelli, A. Di Domenico, B. Dusemund, M. J. Frutos, P. Galtier, D. Gott, U. Gundert-Remy, J.-C. Leblanc, O. Lindtner, P. Moldeus, P. Mosesso, D. Parent-Massin, A. Oskarsson, I. Stankovic, I. Waalkens-Berendsen, R. A. Woutersen, M. Wright, M. Younes, P. Boon, D. Chrysafidis, R. Gürtler, P. Tobback, P. Gergelova, A. M. Rincon and C. Lambré, *EFSA J.*, 2017, **15**(5), e04785.

56 R. J. Bienstock, L. Perera and M. A. Pasquinelli, *Front. Chem.*, 2022, **10**, 880782.

57 C. A. Downs, E. Kramarsky-Winter, R. Segal, J. Fauth, S. Knutson, O. Bronstein, F. R. Ciner, R. Jeger, Y. Lichtenfeld, C. M. Woodley, P. Pennington, K. Cadenas, A. Kushmaro and Y. Loya, *Arch. Environ. Contam. Toxicol.*, 2016, **70**(2), 265–288.

58 C. A. Downs, E. Bishop, M. S. Diaz-Cruz, S. A. Haghshenas, D. Stien, A. M. S. Rodrigues, C. M. Woodley, A. Sunyer-Caldú, S. N. Doust, W. Espero, G. Ward, A. Farhangmehr, S. M. Tabatabaee Samimi, M. J. Risk, P. Lebaron and J. C. DiNardo, *Chemosphere*, 2022, **291**, 132880.

59 C. A. Downs, M. S. Diaz-Cruz, W. T. White, M. Rice, L. Jim, C. Punihaole, M. Dant, K. Gautam, C. M. Woodley, K. O. Walsh, J. Perry, E. M. Downs, L. Bishop, A. Garg, K. King, T. Paltin, E. B. McKinley, A. I. Beers, S. Anbumani and J. Bagshaw, *J. Hazard. Mater.*, 2022, **438**, 129546.

60 M. Aldeghi and C. W. Coley, *Chem. Sci.*, 2022, **13**(35), 10486–10498.

61 G. Ginige, Y. Song, B. C. Olsen, E. J. Luber, C. T. Yavuz and J. M. Buriak, *ACS Appl. Mater. Interfaces*, 2021, **13**(24), 28639–28649.

62 A. Statt, D. C. Kleeblatt and W. F. Reinhart, *Soft Matter*, 2021, **17**(33), 7697–7707.

63 Y. Zhang and X. Xu, *J. Mol. Graph. Model.*, 2021, **103**, 107796.