Leveraging High-throughput Molecular Simulations and Machine Learning for Formulation Design

Alex K. Chew[a,*], Mohammad Atif Faiz Afzal[b], Zachary Kaplan[a], Eric M. Collins[a], Suraj Gattani[a], Mayank Misra[a], Anand Chandrasekaran[a], Karl Leswing[a], Mathew D. Halls[c]

[a] *Schrödinger, Inc., New York 10036, United States*
[b] *Schrödinger, Inc., Portland, Oregon 97204, United States*
[c] *Schrödinger, Inc., San Diego, California 92121, United States*

Abstract

Formulations, or mixtures of chemical ingredients, are ubiquitously found across material science applications, such as themoplastics, consumer packaged goods, and energy storage devices. However, finding formulations with optimal properties is difficult because of the non-obvious connection between the individual ingredient structures and compositions to downstream mixture properties. Computational approaches that could traverse the expansive design space offer a promising solution to finding formulations with improved properties while minimizing the number of experiments. In this work, we generated a large formulation dataset using high-throughput classical molecular dynamics simulations that resulted in more than 30,000 solvent mixtures ranging between pure component to five component systems. We developed three formulation-property relationship approaches to create machine learning models which use the ingredient structure and composition as input to predict a formulation property: formulation descriptor aggregation (FDA), formulation descriptor Set2Set (FDS2S), and formulation graph (FG). We found that FDS2S, a new approach that uses a Set2Set layer to aggregate molecular descriptors of individual ingredients, outperforms all other approaches in accurately predicting density, heat of vaporization (ΔH_{vap}), and enthalpy of mixing (ΔH_m) that were computed from molecular simulations. Feature importance analysis of FDA models reveal that specific substructures are important to predicting these formulation properties, which is useful in the design of formulations to achieve target properties. When leveraging an active learning framework to iteratively suggest the next ingredient and composition to experiment on, we found that formulation-property relationships can identify formulations with the highest property values at least two to three times faster than randomly guessing. The results demonstrate that formulation-property relationships provide valuable insight to suggest the next experiment even when starting from a limited dataset of ~100 examples. Our research demonstrates the utility of high-throughput simulations and machine learning algorithms applied to designing formulations with promising properties, which could broadly accelerate the design of new materials for a wide range of applications, such as improving the performance of liquid electrolytes for batteries, fuel mixtures for oil and gas, solvent additives for perfumes or paints, and more.

Keywords: Formulations, Chemical Mixtures, Classical Molecular Dynamics Simulations, Formulation-Property Relationships, Quantitative Structure-Property Relationships, Machine Learning

[*] Corresponding author
Email address: alex.chew@schrodinger.com (Alex K. Chew)

1. Introduction

Formulations consisting of a mixture of chemical ingredients are crucial to a wide-range of material science applications. These mixtures have multiple chemical ingredients with well-defined compositional information, but their formulation properties are challenging to predict *a priori* because they emerge from non-obvious intermolecular interactions arising between multiple ingredients that heavily depend on both molecular structure and composition. Hence, tuning the chemistry and composition for a desired formulation property is often performed with trial-and-error experiments, which is challenging given the large design space of possible chemical structures and compositions.

As an alternative to experiments, simulating all possible interactions between molecules with classical molecular dynamics (MD) simulations is a promising approach to compute properties of multicomponent systems. For example, MD simulations have been used to study the impact of copolymer blends on polymer properties [1], cosolvents on reactivity [2, 3], and surfactants on cosmetic properties [4]. MD simulations have achieved success in not only accurately capturing experimental trends [5–7], but they have also provided physical insight into the underlying mechanisms that lead to the bulk properties of multicomponent systems, such as phase separation or solvation behavior [2]. Despite significant advances in MD, the utility of MD to simulate formulation systems is limited by the number of atoms in the system, whereby large multicomponent systems with more than $\sim$10 different components may be computationally expensive to simulate but highly prevalent in materials applications like paints, perfumes or fuel [8].

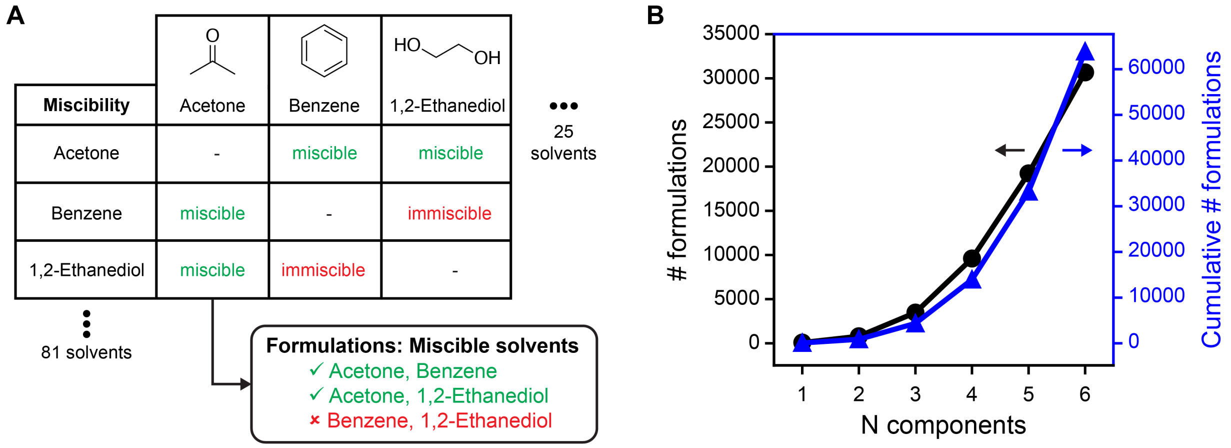
Recent developments in data-driven machine learning modeling that could map chemical structure to bulk properties have shown great promise to speed up chemical discovery, namely quantitative structure-property relationships (QSPR) [9]. QSPR modeling has primarily been focused on single molecule structure-property predictions, where expert-defined cheminformatics descriptors or graph representations are used to train machine learning models [10]. QSPR approaches for single molecules have shown great success in the last decade, especially in the small-molecule drug discovery field [9–11]. However, developing accurate QSPR models for formulation systems have not been well-explored. Recent literature have shown some success on applying machine learning to multicomponent systems, namely the use of various machine learning methods to predict thermodynamic properties [12], variational autoencoders to predict compositions of ingredients [13], and graph neural networks to predict a variety of formulation properties, such as viscosities of binary mixtures [14], battery performance [15, 16], or optical properties of dyes [17, 18]. However, the development of QSPR models for formulation systems (*i.e.* formulation-property relationships) have been largely hindered by the lack of publicly available, comprehensive datasets to evaluate these systems, which makes rigorous benchmarking of formulation-property relationships difficult. Given a sufficiently large formulation dataset, we can begin to tune accurate machine learning models that can handle chemical information aggregated from multiple ingredients and varying compositions.

In this work, we explore QSPR methods for formulation systems to identify the best formulation-property relationships that can accurately predict formulation properties. Given the lack of publicly available experimental data, we generate a representative formulation dataset consisting of ∼30,000 miscible solvent mixtures computed by MD simulations, where ensemble-averaged properties from MD correlate well with experiments. We focus on the capabilities of formulation-property models in predicting three relevant formulation properties, namely packing density, heat of vaporization (ΔH_{vap}), and enthalpy of mixing (ΔH_m). We then apply feature importance analysis tools to identify the top features relevant to formulation-property relationships for each of the formulation property, which provides useful insight into designing formulations for a desired property. Using the extensive formulation dataset generated by MD, we finally leverage an active learning framework to investigate whether formulation-property models can identify the next best formulation to experiment on, starting from a small dataset size of 100 examples. This work highlights the use of high-throughput MD simulations and machine learning models for developing accurate formulation-property relationships, which broadly expands our capabilities to rapidly identify formulations with promising properties for materials applications.

2. Methods

2.1. Formulation dataset: Miscible solvents

Fig. 1A shows the workflow of selecting formulation examples given the miscibility table of 81 solvents that were tabulated against 25 solvents. We first extracted miscibility tables from the CRC handbook to identify pairs of industrially relevant solvents that were miscible with one another from Ref. [19]. Fig. 1A shows an example of binary mixtures selected by using miscibility tables of acetone, benzene, and 1,2-ethanediol. In this example, acetone and benzene are considered a formulation since they are miscible, whereas benzene and 1,2-ethanediol were not considered a formulation since they are immiscible. One limitation of using miscibility tables is that they measure miscibility with equal volumes of two liquids, which does not inform us on whether the mixture is miscible when varying compositions. We further tested whether varying compositions of binary solvent mixtures result in any immiscibilities, and we observed that the majority of the mixtures are miscible based on MD simulations (see Supporting Information Fig. S1 and Fig. S2). For an N-component system, we assumed that if every solvent pair were miscible with one another, then the entire N-component system is assumed to be miscible and considered as a viable formulation. Fig. 1B shows the number of possible unique formulations as we increase the number of components up to six. We arbitrarily selected to study up to five components, which consists of a total of 19,238 unique formulations. By using experimentally derived miscibility tables, we designed a large formulations dataset that consists of miscible solvent mixtures, where homogenous solutions are important in a variety of material science applications such as battery electrolytes, chemical reactivity, and consumer packaged goods.



Miscibility	Acetone	Benzene	1,2-Ethanediol
Acetone	-	miscible	miscible
Benzene	miscible	-	immiscible
1,2-Ethanediol	miscible	immiscible	-

Figure 1: Formulation dataset generated from experimental miscibility tables. **A** Example of three solvents from miscibility tables extracted from Ref. [19]. Pairs of solvents that were labeled "miscible" were used to generate a formulation dataset. A total of 81 solvents were tabulated against 25 solvents for miscibility. **B** Number of unique formulations and cumulative number of unique formulations possible against the number of components using the miscibility table described in (A). The cumulative number of formulations means the cumulative sum of formulations from 1 to N components.

Using the 19,238 unique formulations for up to 5 components, we further varied the composition for binary and ternary systems as summarized in Table 1. We varied the composition for binary mixtures such that each component is varied from 20%, 40%, 50%, 60%, 80%. For ternary mixtures, we selected 60% of one component and 20% of other components, as well as equimolar mixtures. Given the large possibilities of variations for quaternary and quintenary mixtures, only equimolar systems were studied here. In sum, a total of 30,142 formulation examples were studied in this work that span from pure-component systems (N=1) to quinternary systems (N=5).

N components	Compositions	#unique formulations	#examples
1	{100}	81	81

2	{20,80}
	{40,60}
	{50,50}
	{60,40}
	{80,20}
		716	3580

3	{20,20,60}
	{20,60,20}
	{60,20,20}
	{33,33,33}
		2680	10720

4	{25,25,25,25}	6122	6122
5	{20,20,20,20,20}	9639	9639
Total		19238	30142

Table 1: Summary of formulations studied in this work as a function of number of components. Various compositions were varied as shown in brackets. For example, for binary mixtures, {20,80} means 20% of component 1 and 80% of component 2. The number of unique formulations and the total number of examples after variations of compositions are tabulated.

2.2. Classical molecular dynamics simulations

We performed MD simulations for all 30,142 formulation examples to generate the formulation labels necessary to build formulation-property relationships. For all simulations, we used the Schrödinger's Materials Science Suite (MSS) [20], which leverages the Desmond MD engine to rapidly speed up MD computations through GPU acceleration [21]. All molecules were parameterized with the OPLS4 force field [5]. For each system, we first constructed an amorphous simulation cell with approximately 10,000 atoms. The initial density of the system in the amorphous cell structure was 0.5 g/cm^3.

The system was equilibrated with the following procedure: (1) Brownian minimization for 150 ps; (2) a 0.5 ns NVT ensemble (Number of atoms, Volume, and Temperature are conserved) with 2 fs time step at temperature of 500 K and pressure of 1 atm; (3) 1 ns NPT ensemble (Number of atoms, Pressure, and Temperature are conserved) with 2 fs time step at temperature of 400 K and pressure of 1,000 bar; (4) 2 ns NPT ensemble with 2 fs time step at temperature of 300 K and pressure of 1 atm; (5) 5 ns NPT ensemble with 2 fs time step at the 300 K and pressure of 1 atm; (6) 10 ns NPT ensemble with 2 fs time step at temperature of 293 K and pressure of 1 atm. After this equilibration protocol, we take the average cell size of the last 20% of the previous step and subsequently perform 1 ns NVT ensemble with 2 fs time step at a temperature of 293 K. The final production run consists of a 20 ns NVT ensemble with 2 fs time step and temperature of 300 K, where the frames are stored at every 100 ps interval.

We extracted three MD descriptors from the last 10 ns of the production MD simulation: (1) packing density, (2) heat of vaporization ($\mathit{\Delta} H_{vap}$), and (3) enthalpy of mixing ($\mathit{\Delta} H_m$). Density was calculated by dividing the total molecular weight by the simulation cell volume and is reported in g/cm^3.

$\mathit{\Delta} H_{vap}$ is the amount of heat needed to convert some fraction of liquid into vapor. $\mathit{\Delta} H_{vap}$ was calculated from the energy of the periodic unit cell (E_{cell}) minus the sum of the N individual molecules, E_i, averaged over the last 10 ns of the production MD trajectory, as shown in Equation 1.

$$\mathit{\Delta} H_{vap} = \left\langle E_{cell} - \sum_i E_i \right\rangle + RT \tag{1}$$

R is a gas constant with a value of 1.9872036×10^{-3} kcal K^{-1} mol^{-1}, and T is the temperature. $\mathit{\Delta} H_{vap}$ is reported in units of kcal/mol. While measuring $\mathit{\Delta} H_{vap}$ for mixtures is challenging to measure experimentally [22], $\mathit{\Delta} H_{vap}$ has been observed to correlate with temperature-dependent viscosities of pure liquids from MD simulations [23] and experiments [24]. Therefore, $\mathit{\Delta} H_{vap}$ is an informative property that may be correlated to other materials properties.

$\mathit{\Delta} H_m$ is a fundamental thermodynamic property of liquid mixtures that measures the energy released or absorbed upon the mixing of pure components into a single phase in equilibrium. $\mathit{\Delta} H_m$ was calculated using Equation 2 [25].

$$\mathit{\Delta} H_m = \langle E \rangle_m - \sum_i x_i \langle E \rangle_i + PV^E \tag{2}$$

$\langle E \rangle_m$ is the ensemble average cohesion energy of the mixture, x_i is the mole fraction of component i, $\langle E \rangle_i$ is the ensemble average cohesion energy of pure component i, P is the pressure, and V^E is the excess volume of the mixture. Previous work have use kinetic and/or potential energies to estimate $\mathit{\Delta} H_m$ [25, 26], but we observed that cohesion energy performed slightly better in agreeing with experiments (results are not explicitly shown here). V^E is calculated using Equation 3.

$$V^E = \langle V \rangle_m - \sum_i x_i \langle V \rangle_i \tag{3}$$

$\langle V \rangle_m$ is the ensemble average volume of the mixture, and $\langle V \rangle_i$ is the ensemble average volume of pure component i. $\mathit{\Delta} H_m$ is reported in units of kJ/mol. We treat these three MD descriptors as relevant formulation labels that are applicable to material science applications. For example, density is an important property for battery applications since it dictates the battery weight and charge mobility; $\mathit{\Delta} H_{vap}$ is a property that effectively measures the cohesion energy of a liquid and has been previously observed to correlate with viscosity [23]; and, $\mathit{\Delta} H_m$ is important for process design that dictates properties, such as solubility and phase stability.

2.3. Formulation-property relationships

All formulation-property relationships were built using the DeepAutoQSAR framework, Schrödinger's automated molecular property prediction engine [27, 28]. In DeepAutoQSAR, feature and model hyperparameter selection are iteratively improved by Bayesian optimization based on the model performance on the previous training cycle. This work extends the DeepAutoQSAR workflow to be able to encode formulations as inputs, where multiple molecules with compositions are inputted rather than only single molecule property predictions. We focused on formulation-property relationships that have the following ideal characteristics: (1) composition must be accounted for in the model such that variations in compositions impact property predictions; (2) models must be permutationally invariant, such that changing the order of input molecules and compositions do not change the prediction output; and, (3) models are flexible to the number of components, such that a model trained with binary mixtures can be used to predict ternary mixtures, quarternary mixtures, and so on. These model characteristics are important for designing formulations because composition is crucial to a formulation, ingredients can be inputted in a random order, and the inclusion or removal of particular ingredients is commonly evaluated to measure the impact of individual ingredients to formulation properties.

Fig. 2 summarizes three different approaches for developing formulation-property relationships that satisfies the characteristics of an ideal model. Fig. 2A shows the formulation descriptor aggregation (FDA) approach where individual molecules are featurized, weighted by their corresponding compositions, then aggregated by performing a variety of statistical metrics like computing the mean, standard deviation, minimum, maximum, and median. These aggregated features are considered as formulation descriptors, which are then passed as inputs into ML models to predict formulation property. By aggregating with statistical approaches, the formulation descriptor captures the distribution of molecular properties of individual ingredients, which would be useful for property prediction. The FDA approach is analogous to matminer featurizers that perform statistical operations, such as averaging and standard deviation, to characterize inorganic materials by aggregating features from individual atomic types [29].

Fig. 2B shows a similar descriptor-based approach as FDA, but instead of aggregating with statistical approaches, the compositionally weighted descriptors are passed into a Set2Set algorithm [30] to create a formulation descriptor vector (FDS2S). The Set2Set operator uses a combination of long short-term memory networks to process sequential data and softmax function as an attention layer to aggreate multiple arrays coming from multiple molecules into a single array [30]. Set2Set outputs the same array even when the order of the input array is changed, thus satisfying the requirement of permutation invariance for an ideal formulation-property model. The final array from the Set2Set layer is then passed to a fully connected layer to predict the formulation property. The usefulness of Set2Set as a way to aggregate information has been seen in several previous works, such as aggregation of reactant or product information to predict bond disassociation energies [31] or hydrolysis energies [32].

Fig. 2C shows a graph-based representation approach (FG), where atoms are nodes and bonds are edges. Each node vector consists of 75 atomic features and the composition of the ingredient. For each node, graph convolution operators aggregate information from the neighboring nodes and output a new atomic vector based on message passing across the molecular graph. The final learned atomic features are then outputted to a readout layer, which are then input to a fully connected neural network to predict the formulation property. Previous work have shown success in using graph-based representations for predicting viscosity of binary mixtures [14] and battery performance of electrolyte systems [15].

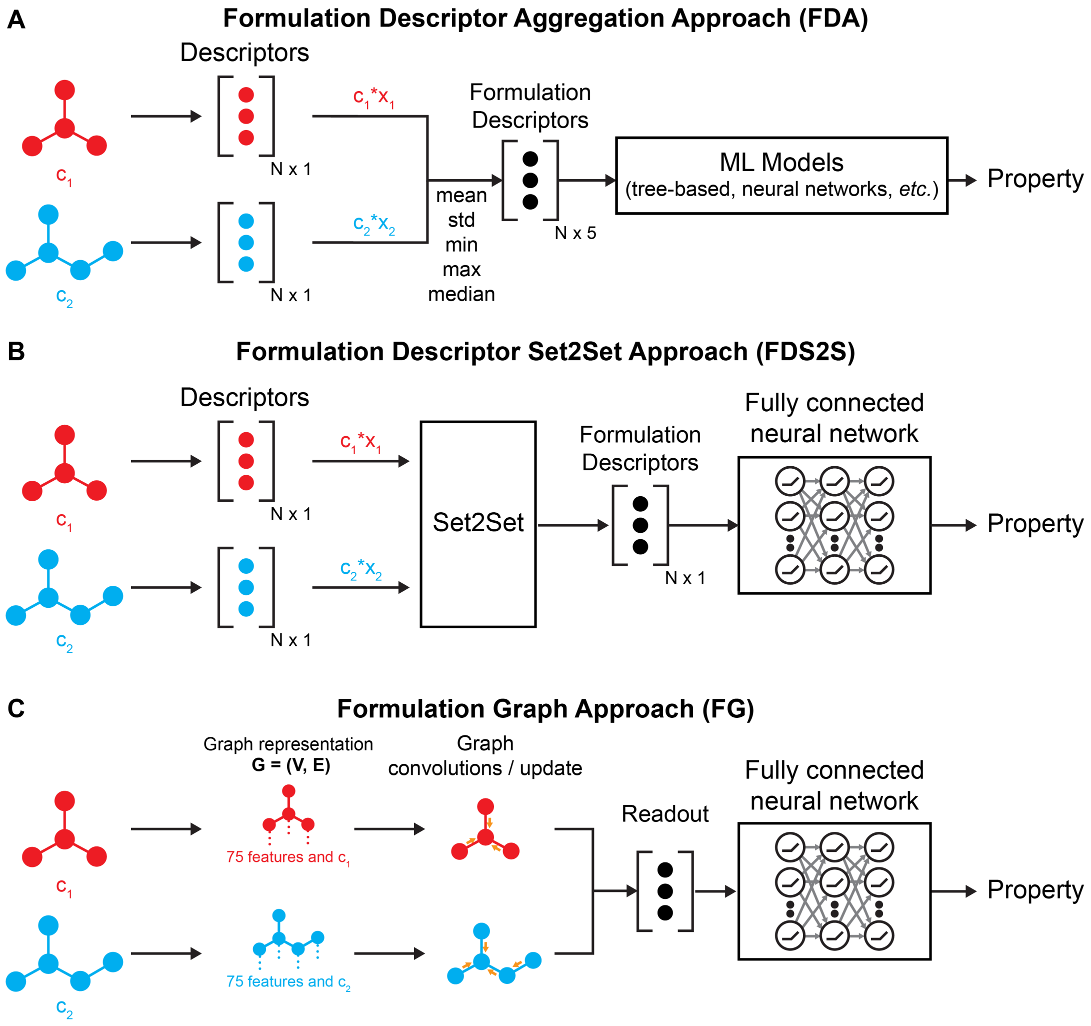


Figure 2: Schematic of formulation-property relationship approaches. **A** Formulation descriptor aggregation approach (FDA) where two molecules are featurized to generate molecular descriptors that are compositionally weighted, then aggregated by computing the mean, standard deviation (std), minimum (min), maximum (max), and median values. These aggregated features are considered as formulation descriptors that are passed into machine learning algorithms to predict formulation properties. **B** Formulation descriptor Set2Set approach (FDS2S) where two molecules are featurized to generate molecular descriptors that are compositionally weighted, then these descriptors are aggregated using a Set2Set algorithm, and finally the aggregated descriptors are passed into a fully connected neural network to predict formulation properties. **C** Formulation graph approach (FG) where two molecules a represented as graphs (G) consisting of atoms as nodes (V) and bonds as edges (E). For each molecule, 75 atomic features and composition are used in the node vector. Graph convolutions and update operations are performed, followed by a readout layer and a fully connected neural network to predict formulation properties.

For descriptor-based approaches (*i.e.* FDA and FDS2S), four distinct molecular featurization approaches were evaluated: (1) 200 RDKit descriptors; (2) Morgan fingerprints with a size of $\sim$500-2,060 and radius of $\sim$2-4; (3) 167-bit MACCS keys, which are 2D structure fingerprints commonly used to measure molecular similarity or virtual screening [33]; and, (4) 132 matminer descriptors. Featurization for RDKit, Morgan fingerprints, and MACCS keys were implemented using the rdkit package (Version 2023.9.5) [34], whereas matminer descriptors were implemented using the matminer package (Version 0.9.0) [29]. All features were preprocessed with the following procedure: (1) constant features with variance of zero were removed; (2) correlated features with Pearson's r greater than or equal to 0.90 were removed; and, (3) features were standardized by subtracting the mean and dividing by the standard deviation. For the FDA approach, the use of 200 RDKit descriptors as a featurizer were omitted because of the poor generalizability to new formulations for specific data splits, which is likely because these descriptors are molecular size dependent (*e.g.* molecular weight). For the FG approach, 75 atomic features were used to featurize each of the heavy atoms. Atomic featurizations include one-hot encodings of atomic number, implicit valence, formal charge, atomic degree, number of radial electrons, hybridization, and aromaticity [28]. The composition of the molecule was added as the 76th atomic feature to all nodes. Node features were preprocessed by removing correlated features with Pearson's r greater than or equal to 0.90 and non-binary features were standardized by subtracting the mean and dividing by the standard deviation.

For the FDA approach, four ML algorithms were tested: elastic net, support vector regression, extreme gradient boosting (XGBoost) [35], and fully connected neural network. For the FDS2S approach, only a model with the Set2Set layer [30] and fully connected neural network was used. For the FG approach, ten graph-based approaches were evaluated: Graph Convolution Neural Network (GCN) [36], Pytorch version of GCN (TorchGraphConv) [37], TopK [38], GraphSAGE [39], Graph Isomorphism Network (GIN) [40], Self-Attention Graph Pooling (SAGPool) [41], EdgePool [42], GlobalAttention [40], Set2Set [30], and SortPool [43]. Different GNN models differ slightly by how they aggregate information based on successes from previous literature [40, 42]. Elastic net and support vector regression were implemented using the scikit-learn package (Version 1.2.1)[44]. XGBoost was implemented with the xgboost package (Version 1.7.4) [35]. Fully connected neural networks and graph-based models were trained with PyTorch (Version 2.0.1) [45]. The details of each ML algorithm and hyperparameters are summarized in Ref. [28]. All formulation-property training and prediction workflows are available as the "Formulation Machine Learning" panel within the Schrödinger's Materials Science Suite, Release 2024-2 [46].

2.4. Evaluation of formulation-property models

Since the formulation dataset contains multiple entries with the same set of molecules with different compositions, we implemented an out-of-sample approach for data splitting, where unique formulations are iteratively introduced to the training set until it reaches 90% of the dataset and the remaining 10% of the data is placed in the testing set. Previous studies have emphasized that out-of-sample splitting is a better approach to measure model accuracy as compared to random splitting from an application standpoint because the model performance from random splitting may lead to over-optimistic model performance for datasets with repeated molecules where the same molecule could appear in both train and test sets [47]. A learning curve was generated by setting aside 10% of the 30,142 formulation example dataset as the test set, where the test set is explicitly selected to be unique formulations that are not observed in the set used for training. Portions of the remaining 90% of the 30,142 formulation example dataset was used to train formulation-property relationships, where the trained model was then used to evaluate the left-out test set. To alleviate possible biases of the random, out-of-sample train/test split, this procedure is repeated a total of three times with different random seeds, where the average test set performance is reported and the uncertainty is estimated by computing the standard deviation of the three seeds.

For model training, all featurizers and model hyperparameters are selected using DeepAutoQSAR's Bayesian optimization approach [28]. In this approach, the training set is partitioned into five sets used for five-fold cross validation (5-CV). For each of the five folds, one set is left-out as the validation set and the remaining sets are used to train the model; this procedure is repeated five times until all of the data instances are within the left-out set exactly once. DeepAutoQSAR uses the performance of 5-CV to evaluate the model's ability to generalize to new examples, which is used by the Bayesian optimization algorithm to select the next best featurizer and model hyperparameters to test next. A total of 20 iterations of model training cycles were performed, and the three best-performing models with the highest 5-CV score are selected as the final ensemble model. For training sizes larger then 10,000 examples, the training set was randomly downsampled to 10,000 examples for hyperparameter tuning to improve computational efficiency, and the three best-performing models were re-trained with the entire training set as the final ensemble model. The best hyperparameters when training the formulation-property models with 90% of the dataset are tabulated in Table S2 of the Supporting Information.

2.5. Feature importance of formulation-property models

Feature importance of formulation-property models were only applied to the FDA approach because pre-defined descriptors are easier to interpret than graph-based representations. Given a trained formulation-property model, feature importance was calculated using the SHapley Additive exPLanations (SHAP) approach (shap package, Version 0.42.1), which is a game theory approach to quantify the contributions of single players in a collaborative game [48, 49]. Shapley values measure the impact of a formulation descriptor to an output property by including or excluding the descriptor across a set of instances. For all SHAP calculations, we use the test set instances to measure descriptor importance. The average magnitude of Shapley values is reported (*i.e.* Mean |SHAP|), and the sign of the importance is determined by computing the Pearson's r correlation coefficient between the Shapley and descriptor values. Positive Pearson's r between Shapley and descriptor values indicate that the feature positively contributes to the output property, whereas negative Pearson's r indicates the converse. Additional details about the SHAP method could be found in previous literature [9, 50, 51]. For an ensemble of models, the aggregation of SHAP values are used to compute the Mean |SHAP|.

2.6. Active learning with formulation-property models

Active learning is an iterative supervised learning to guide materials design, where starting with a small dataset, a machine learning model is trained and evaluated on a large pool of examples to suggest the next candidates to measure propertes; the cycle is repeated until the desired property values are obtained. The benefit of an active learning approach is that it leverages data-driven techniques to make informed decisions on the next best candidates rather than random guessing. The suggestion of next candidates at each iteration are determined by the acquisition function (α), which often tries to balance between exploitation (sampling a space where a target property is achieved) and exploration (sampling a space where prediction uncertainty is high). We evaluated four acquisition functions that have been studied in previous literature [52–54], where $\mu(x)$ is the average prediction of sample x, $\sigma(x)$ is the prediction uncertainty of sample x (estimated by computing the standard deviation of the predictions from the individual models of the ensemble):

1. **Expected improvement (EI)** acquisition function (α_{EI}) select samples based on balancing both exploration and exploitation described in Equation 4 and 5.

$$\alpha_{EI} = z\mathit{\Phi}(z) + \sigma(x)\phi(z) \tag{4}$$

$$z = \mu(x) - f(x^*) - \xi \tag{5}$$

   where $\mathit{\Phi}$ is the normalized cumulative distribution function, ϕ is the normalized probability distribution function, $f(x^*)$ is the best performing prediction relative to the target objective, and ξ is the arbitrary constant that dictates the extent of exploration (ξ is set as zero for this work).

2. **Greedy** acquisition function (α_{greedy}) selects samples based on maximizing the target objective described in Equation 6.

$$\alpha_{greedy} = \max \mu(x) \tag{6}$$

3. **Most uncertain** acquisition function ($\alpha_{uncertain}$) selects samples based on the highest prediction uncertainty described in Equation 7.

$$\alpha_{uncertain} = \max \sigma(x) \tag{7}$$

4. **Random** acquisition function selects samples randomly by assigning a random number from a uniform distribution to each sample.

The performance of formulation-property relationships and these acquisition functions were evaluated by setting aside 10% of the 30,142 formulation example dataset as the test set, which were explicitly selected to be unique formulations that are not sampled by the active learning workflow. For each iteration of active learning, the performance of the test set is measured to evaluate the models' ability to generalize to unseen formulations. Of the remaining 90% data, an initial batch of 100 examples were randomly selected as the training set. For each iteration, formulation-property models were trained, used to evaluate the left-out test set, and used to determine the next candidates to include in the training set based on the acquisition function. The active learning cycle was repeated with increments of 100 examples until the training size reached 2,000 examples. The active learning performance was evaluated by computing the 10% left-out test set coefficient of determination (R^2) as a measure of model generalizability and by computing the ability of the models to recapture the top 5% of structures in the training set as a function of training size. For each acquisition function, three individual runs were performed based on three random seeds to accurately measure the active learning performance. The reported performance is the average of the random seeds, and the uncertainty is measured by computing the standard deviation of the performance of each seed. We arbitrarily selected to maximize all formulation properties when evaluating the performance of formulation-property models in an active learning framework. For each training iteration, we enabled the DeepAutoQSAR framework to choose any of the three formulation-property relationships from Fig. 2. For featurizers, we enabled MACCS keys, Morgan fingerprint, and graph representations. For models, we enabled neural network models, Set2Set models, or GlobalAttention graph-based models [40]. These featurizers and models were selected based on the best hyperparameters when trained with 90% of the dataset (see Table S2 in the Supporting Information). A total of 20 iterations of model training cycles were performed, and the three best-performing models with the highest 5-CV score are selected as the final ensemble model.

3. Results and Discussion

3.1. Generating large formulation dataset with classical molecular dynamics simulations

We first validated whether simulation-derived properties can accurately capture experimental trends for industrially relevant solvents. Fig. 3A shows an example of acetone and benzene that are equally weighted and simulated with MD to compute formulation properties. The simulation snapshot from Fig. 3A shows a well-mixed system of acetone and benzene, which is consistent with the experimental miscibility table in Fig. 1A. Fig. 3B shows the correlation coefficient (R^2) between simulation-derived and experimental properties for density, ΔH_{vap}, and ΔH_m. For all formulation properties, we observe good agreement between simulation-derived and experimental properties with a $R^2 \geq 0.84$. Fig. 3C-E shows the parity plot between simulation-derived and experimental properties. For density (Fig. 3C), we compared the packing density of eleven pure solvents and observe a strong agreement against experiments with a R^2 of 0.98 and root-mean-squared error (RMSE) of $\sim$15.4 g/cm^3. Similarly, we observe a strong correlation between MD simulations and experiments for ΔH_{vap} when comparing 34 pure solvents (Fig. 3D), which acheived an R^2 of 0.97 and RMSE of 3.4 kcal/mol. Density and heat of vaporization are expected to be well-captured from MD simulations since the OPLS4 forcefield is parameterized to accurately predict these properties [5]; hence, the results in Fig. 3C and 3D are consistent with the literature in that these two properties are accurately predicted with MD simulations [5–7]. On the other hand, ΔH_m is not used to parameterize the OPLS4 forcefield, but ΔH_m has shown good agreement between experiments and MD simulations for a variety of solvents, such as nonpolar-nonpolar mixtures (*e.g.* benzene and cyclohexane) and nonpolar-polar mixtures (*e.g.* benzene and ethanol) [25]. Fig. 3E shows that simulation-derived ΔH_m captures experimental trends for 53 binary mixture examples using the simulation protocol in this work. Given that the simulation-derived properties correlate with experiments for density, ΔH_{vap}, and ΔH_m, we validated that MD simulations can accurately capture formulation properties for solvent systems studied in this work.

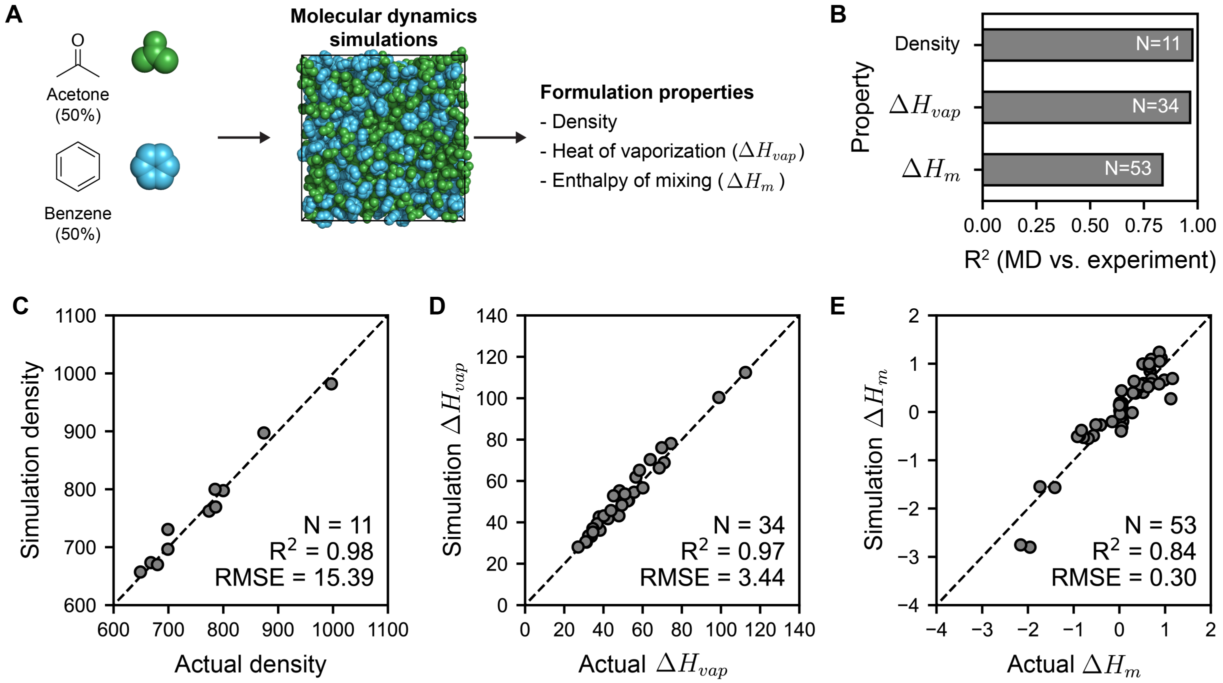


Figure 3: Generating formulation labels using classical molecular dynamics (MD) simulations and validating them against experiments. **A** Workflow to compute formulation properties by adding a 50 wt% acetone and 50 wt% benzene mixture into a MD simulation. Formulation properties are computed using the last 10 ns of a production MD run. **B** Coefficient of determination (R^2) between MD simulation and experimental values for density, heat of vaporization ($\mathit{\Delta H}_{vap}$), and enthalpy of mixing ($\mathit{\Delta H}_m$). N denotes the number of datapoints used for each validation. **C** Simulation-derived versus experimental density for eleven pure solvent examples. **D** Simulation-derived versus experimental $\mathit{\Delta H}_{vap}$ for 34 pure component examples. Experimental densities and $\mathit{\Delta H}_{vap}$ were taken from the CRC handbook [19]. **E** Simulated versus experimental enthalpy of mixing for 54 binary mixture examples. Experimental enthalpy of mixing values were extracted from Ref. [25]. All scatter plots contain coefficient of determination (R^2) and root-mean-squared error (RMSE) between simulation and actual values in the lower right corner. A diagonal gray dashed line is shown as a visual guide. The examples used to compare the formulation labels between MD simulations and experiments are tabulated in Table S1 of the Supporting Information.

Since MD simulations can accurately capture experiment trends, we then used MD simulations to generate a large formulation dataset that is useful to benchmark formulation-property relationships. Using the miscibility table to identify miscible solvent systems ranging from pure component systems ($N = 1$) to quinternary systems ($N = 5$) as described in Fig. 1, we performed 30,142 MD simulations and extracted the density, $\mathit{\Delta H}_{vap}$, and $\mathit{\Delta H}_m$ from the production simulations (see the Methods section for simulation details). Fig. 4 shows the box and whisker plot of density, $\mathit{\Delta H}_{vap}$, and $\mathit{\Delta H}_m$ computed from MD simulations as a function of number of components. Fig. 4A and Fig. 4B shows that as the number of components increase, the distribution of density and $\mathit{\Delta H}_{vap}$ are more narrow as compared to pure component systems ($N = 1$). These results show that pure component systems have a large range of properties as compared to when mixing the individual components, and mixtures of solvents can be used to fine-tune properties to highly specific values that is not possible when only using pure component systems. Similar to density and $\mathit{\Delta H}_{vap}$, Fig. 4C shows that increasing number of components results in narrower ranges for $\mathit{\Delta H}_m$.

However, ΔH_m differs from the other two properties in that pure component systems will have $\Delta H_m = 0$ because ΔH_m of a mixture is relative to its corresponding pure component systems. Hence, binary systems ($N = 2$) have the largest range of ΔH_m values. Since ΔH_m is a relative mixture property, it may be a challenging property to predict with formulation-property relationships as the model will need to learn differences between the mixture and its individual components. We use the 30,142 formulations with the three property labels from MD simulations to evaluate whether the formulation-property approaches in Fig. 2 can be used to create accurate models.

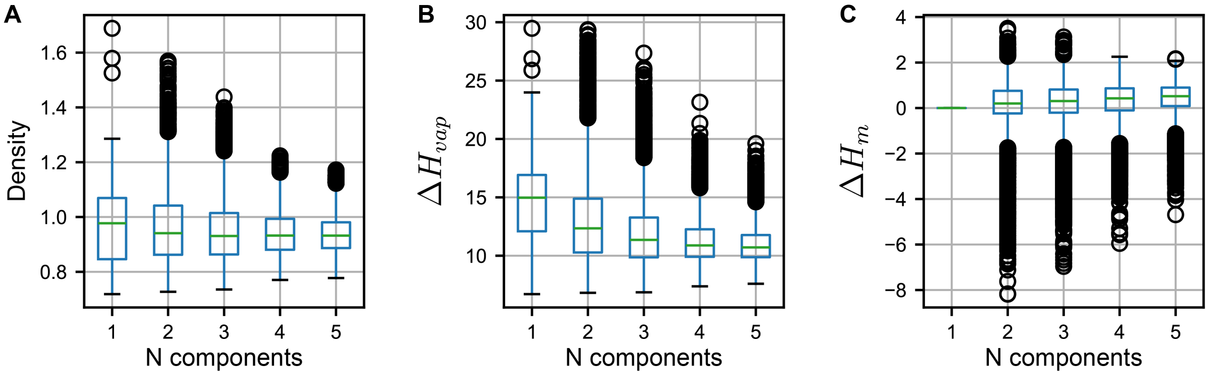


Figure 4: Distribution of the formulation labels from classical molecular dynamics simulations. Box and whisker plot between formulation labels versus number of components are shown for (**A**) density, (**B**) heat of vaporization (ΔH_{vap}), and (**C**) enthalpy of mixing (ΔH_m). Gray grid lines are shown as visual guides.

3.2. Performance of formulation-property models

We next evaluate the performance of the different formulation-structure approaches (Fig. 2) on predicting the three formulation properties extracted from MD simulations (Fig. 4). The performance of each formulation-property approach is measured by using a learning curve, where a machine learning algorithm is iteratively trained with incrementally increasing training sizes to determine its prediction accuracy on a left-out test set as a function of training set size. An ideal formulation-property model should be able to accurately predict formulation properties at both small (∼100 examples) and large (>1000 examples) dataset sizes, especially since many formulation datasets are often data limited. For example, a recent study had fewer than 200 electrolyte formulations that were experimentally available to evaluate machine learning approaches on predicting battery charging efficiencies [15], which makes benchmarking data-driven approaches for formulations challenging. By using MD simulations to generate formulation labels, we can rigorously analyze the performance of formulation-property relationships at both small and large dataset sizes, which would be useful to identify formulation-property approaches that are accurate for a broad range of training sizes.

Fig. 5A-C shows the learning curve performance of FDA, FDS2S, and FG models when predicting density, ΔH_{vap}, and ΔH_m. Each learning curve shows the test set R^2 as a function of training set size. When the target property is density (Fig. 5A), all formulation-property models achieve test set $R^2 \sim 0.90$ when >500 training examples are available, which demonstrates that the formulation-property models can accurately predict density with relatively small dataset sizes. When the training size is less than 100, FDS2S models outperform FDA and FG approaches in predicting the test set density. Of the three target properties, density is the easiest property for formulation-property models to predict, which may be due to its general monotonic behavior as a function of composition for most binary mixtures [25]. Fig. 5B shows that formulation-property models can also accurately capture ΔH_{vap} with a test set $R^2 \geq 0.80$ when >500 training examples are available. Interestingly, FG models struggle to predict ΔH_{vap} when the training size is less than 200, whereas descriptor-based models (FDA and FDS2S) achieve test set $R^2 \geq 0.60$ at this limited data region. The poor prediction accuracy of FG models is likely due to poor representations generated when using graph convolution neural networks when limited data is available. Pre-defined descriptors that can better represent the material at the small data scale have been shown to outperform graph-based models, where graph models that automatically learn molecular representations through convolutional operations require sufficient amount of training data to obtain informative molecular features [9, 23]. Similar to density, FDS2S outperforms the other models in predicting ΔH_{vap} across all training sizes. Fig. 5C shows that formulation-property models generally struggle to predict ΔH_m until the training size is at least ~5,000 examples, which achieve a test set $R^2 \geq 0.80$. FDS2S performs the best in predicting ΔH_m for majority of the training sizes. At the large training sizes, FDS2S and FG models outperform FDA models, which highlights the strength of deep neural networks and learned representations at the large data scale when predicting complex properties. ΔH_m is a relative property of a mixture to pure component systems, which adds to the complexity of creating accurate formulation-property relationship as differences of the mixtures to pure component systems are not explicitly defined in formulation-property relationships. One possible way to improve the predictions to ΔH_m is to encode descriptor differences between multiple species to improve the predictions of relative properties, such as taking differences between reactant and product feature space to improve the prediction of bond dissociation energies [31] or hydrolysis energies [32], which is a subject of future work.

Given that FDS2S demonstrated high test set R^2 for all formulation properties in both small and large training sizes, we further analyzed the performance of FDS2S on the test set. Fig. 5D-F shows the parity plot between predicted and actual values for density, ΔH_{vap}, and ΔH_m of the left-out test set when FDS2S models are trained with 90% of the data (*i.e.* training size of 27,127). Fig. 5D and Fig. 5E shows that formulation-property models can accurately predict density and ΔH_{vap} for new formulation examples with test set R^2 close to unity. Furthermore, Fig. 5E shows that properties like ΔH_m, which are challenging to predict, can also be accurately predicted with a test set R^2 of 0.96 when a large number of data points are available. The results in Fig. 5 demonstrate that the FDS2S approach achieves high accuracy in predicting all the three formulation properties, and the FDS2S approach ranks higher than the FDA and FG approach in consistently creating accurate formulation-property models for both small and large dataset sizes. From the best of our knowledge, the FDS2S approach to create accurate formulation-property models have not yet been reported in the literature, and the results from Fig. 5 suggests that FDS2S is a promising approach to leverage the strengths of traditional descriptor-based approaches (*e.g.* FDA) at the small data scale and strengths of graph-based approaches (*e.g.* FG) at the large data scale to creating accurate formulation-property models regardless of dataset size.

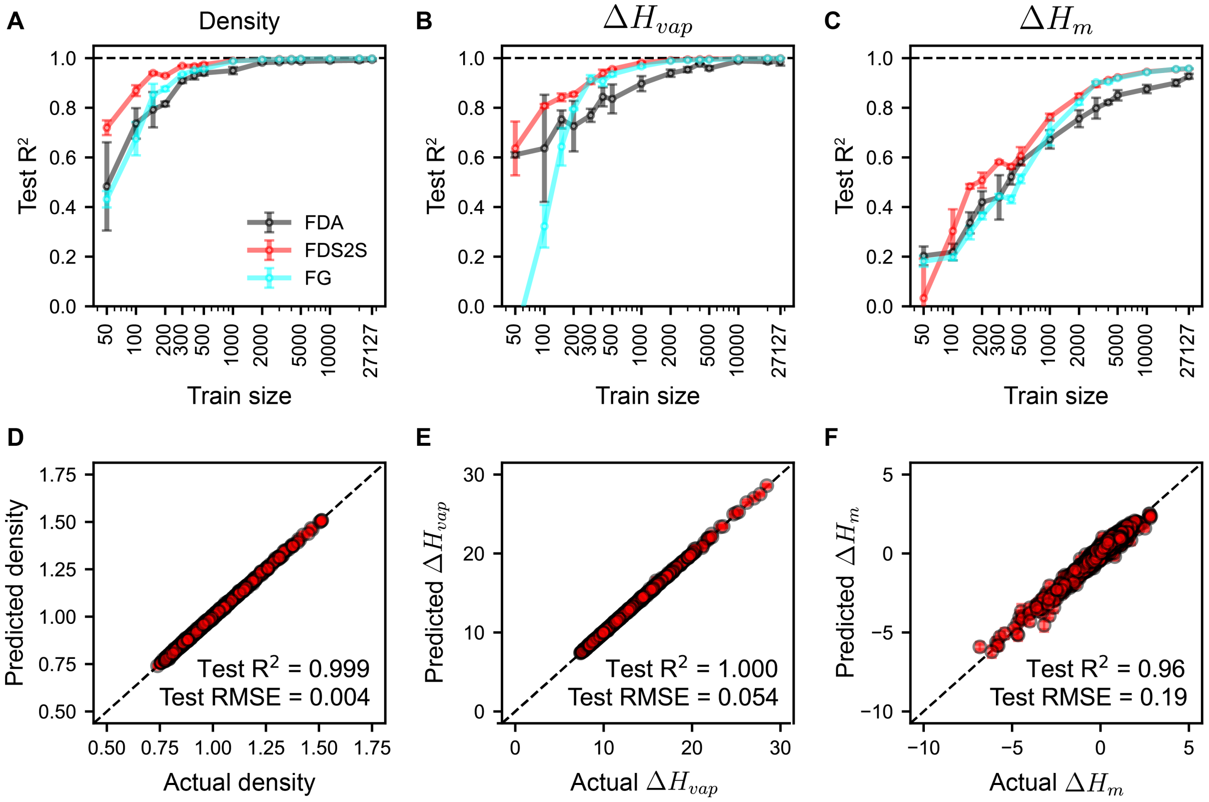


Figure 5: Performance of formulation-property relationships. Learning curve showing the left-out test set coefficient of determination (R^2) as a function of training size when formulation-property models are trained to predict (**A**) density, (**B**) heat of vaporization (ΔH_{vap}), and (**C**) enthalpy of mixing (ΔH_m). The average test set R^2 of three independent runs are shown, and the uncertainty is estimated by computing the standard deviation of the individual runs. Dashed black line is drawn at test set R^2 of 1 as a visual guide. The parity plots between predicted and actual values of the test set when FDS2S models are trained with 90% of the data (27,127 examples) are shown for (**D**) density, (**E**) ΔH_{vap}, and (**F**) ΔH_m. For parity plots, the test set R^2 and root-mean-squared error performance is shown in the bottom right and a dashed black diagonal line is drawn as a visual guide.

3.3. Feature importance of formulation-property models

Since machine learning models achieved a high test set accuracy ($R^2 \geq 0.90$) when trained with 90% of the data, we next sought to identify the top relevant features that were useful to predict density, ΔH_{vap}, and ΔH_m. Of the formulation-property approaches shown in Fig 2, the FDA approach is the most straightforward to perform feature importance analysis because predefined descriptors are more easy to interpret than graph-based representations. The FDA approach perform similarly to FDS2S and FG approaches at 90% of the training data (see training size of 27,127 in Fig. 5A-C), hence we would expect the top molecular descriptors relevant to formulation properties from the FDA approach might be similar to the FDS2S and FG approaches. We selected to use the SHAP approach to analyze the top features for FDA models because the SHAP approach is model agnositic that enables the evaluation of feature importance across different machine learning algorithms and have been observed to capture relevant top features in previous works [23, 50, 51, 56, 57] (see Methods section for details on how SHAP is computed).

]

Fig. 6 shows the top three descriptors using the SHAP approach for FDA models when trained to predict density, ΔH_{vap}, and ΔH_m; example structures of individual solvent ingredients are highlighted to the right of each descriptor. Fig. 6A shows that MACCS keys features were the most relevant features to accurate predictions of density. The mean MACCS keys of 160 and 114 contribute negatively to density, where the removal of low molecular weight methyl and ethyl groups lead to an increase in density. Conversely, the mean MACCS key of 107 contributes positively to density, which means that inclusion of high atomic weight halogen elements lead to an increase in density.

Fig. 6B shows that Morgan fingerprints were the most useful features to accurately predicting ΔH_{vap}. The mean of the top Morgan fingerprints are all positively correlated with ΔH_{vap}, namely the inclusion of benzene rings, hydroxyl groups, and methylene units. ΔH_{vap} is related to the cohesion energy of a solution; hence, favorable interaction energies between molecules in a mixture would typically lead to high ΔH_{vap} values. Therefore, the inclusion of benzene rings may lead to π-π stacking, which is well-known to be a favorable interaction in the literature [58]. Furthermore, the inclusion of hydroxyl groups lead to favorable hydrogen bonding, and the inclusion of long methylene chains could lead to favorable nonpolar interactions [59]. Interestingly, Morgan fingerprint of index 46 with fingerprint size 952 (mean-MorganFingerprint_46_952) shows a bit-collision between benzene and hydroxyl groups, where the bit-fingerprint is set to unity for multiple atomic environments. While bit-collisions lead to information loss of distinct atomic environments, the importance of hydroxyl groups are re-iterated in mean-MorganFingerprint_536_1050, which suggests that bit-collisions did not significantly impact the interpretability of top features. In sum, ΔH_{vap} can be increased by including ingredients with benzene groups, hydroxyl groups, or methylene units.

Similar to ΔH_{vap}, Fig. 6C shows that Morgan fingerprints were top features relevant to predicting ΔH_m. Interestingly, all top features relevant to ΔH_m are nitrogen containing compounds, and they all contribute negatively to ΔH_m. Previous literature have reported mixtures with nitrogen containing compounds, such as diethylamine and ethanol, have negative ΔH_m values with increasing diethylamine content [25], which is consistent with the top features in Fig. 6C. Therefore, ΔH_m can be potentially tuned by including or removal of ingredients with nitrogen-containing groups. The results in Fig. 6 demonstrate that top features related to a property can be extracted from formulation-property models, which can be used to fine-tune the selection of ingredients that satisfy a desired property criteria.

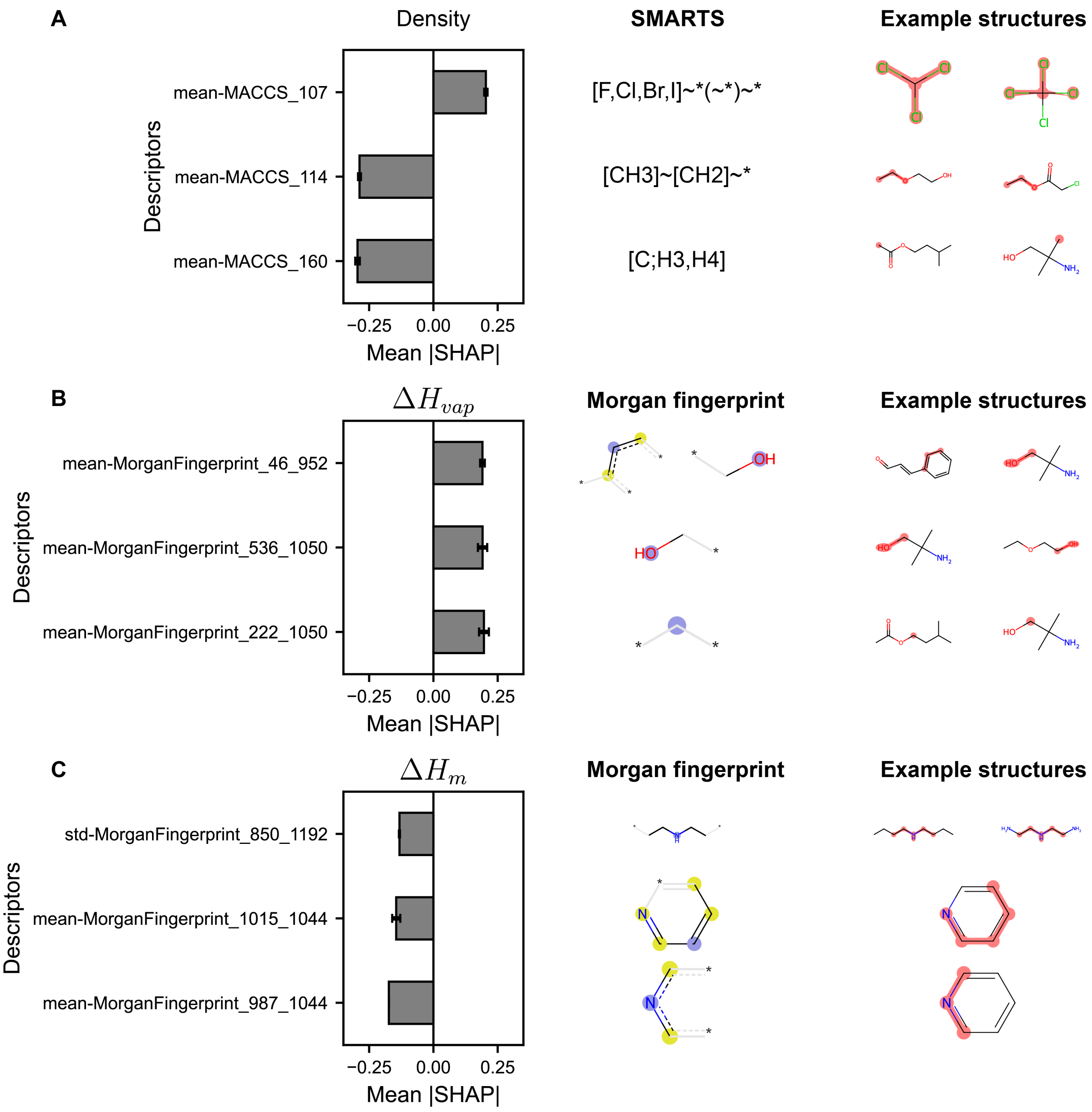


Figure 6: Feature importance from FDA models. Top three most important features measured as the average magnitude of SHapley Additive exPLanations (SHAP) values (*i.e.* Mean |SHAP|) are shown for FDA models trained with 90% of the 30,142 formulation examples to predict (**A**) density, (**B**) heat of vaporization (ΔH_{vap}), and (**C**) enthalpy of mixing (ΔH_m). Positive Mean |SHAP| indicates that the descriptor positively contributes to the formulation property, whereas negative Mean |SHAP| indicates the converse. The average Mean |SHAP| of three models of an ensemble is reported and the uncertainty is estimated by the computing standard deviation of the Mean |SHAP| values. For descriptors, prefixes with "mean" and "std" means that the compositionally weighted descriptor of individual ingredients was aggregated with average and standard deviation operations, respectively. For MACCS keys descriptors, the index of the MACCS key is shown in the right-most value (*e.g.* mean-MACCS_107 means the 107th MACCS key). For Morgan fingeprint descriptors, the index and total length of the bit-fingerprint is shown as the two right-most values (*e.g.* mean-MorganFingerprint_46_952 means a Morgan fingerprint index of 46 with a fingerprint size of 952). SMARTS pattern for MACCS keys, Morgan fingerprints, and example structures with red highlighted patterns are illustrated to the right of the SHAP plots.

3.4. Active learning using formulation-property models

While formulation-property models are highly accurate with a large amount of data and can be subsequently used to extract important features relevant to a property, formulations design is often performed at the small data scale (∼100 examples). Hence, we next evaluated whether formulation-property models are useful for identifying top candiates at the small data scale starting from 100 examples using an active learning approach. The typical approach for active learning is by using a surrogate model (*i.e.* a machine learning model) to train on a small subset of data and predict on a large pool of candidates; then, based on the predictions of the model, suggest the best candidates to evaluate in the next experiment. After the best candidates are evaluated, they are added as part of the training data, then the loop is repeated a set number of iterations until the desired property criteria is reached. The selection of best candidates from the machine learning predictions is determined based on the acquisition function. We evaluate four acquisition functions: expected improvement, greedy, most uncertain, and random acquisition functions (see Methods for details).

Fig. 7 shows the performance of using formulation-property models in an active learning framework to identify formulations with the highest density, ΔH_{vap}, and ΔH_m. Fig. 7A-C shows the R^2 performance of formulation-property models on a 10% left out test set as a function of training size when using four distinct acquisition functions. Fig. 7D-F shows the percentage of formulations within the top 5% of density, ΔH_{vap}, or ΔH_m that were selected to be in the training set during the active learning iterations. For density as a target property, Fig. 7A shows that all acquisition functions result in a test set R^2 of ∼0.90 when the formulation-property model with less than 500 examples. The greedy acquisition function has a lower test set R^2 as compared to the other acquisition functions, suggesting that the greedy acquistion function results in models that are not as generalizable as compared to random selection. However, even though the greedy acquisition function results in less accurate models, Fig. 7D shows that the greedy acquisition function captures close to 90% of the top 5% density values after the training sizes reach ∼1,500 examples. Conversely, the expected improvement and most uncertain acquisition functions only achieve ∼20% of the top density candidates at the same training size. The random selection acquisition function is expected to be the worst with less than 5% of the top density values identified. At 2,000 examples, the greedy acquisition function identified formulations with the highest density values 14-folds higher than when randomly selecting formulations.

Similar to density as a target property, Fig. 7B shows that all acquisition functions result in ΔH_{vap} models that achieve a test set R^2 of ∼0.90 when the training set contains 500 examples, and the greedy acquisition function generally has lower test set R^2 as compared to the other acquisition functions. Interestingly, Fig. 7E shows that greedy, expected improvement, and most uncertain perform similarly in identifying formulations with the top 5% ΔH_{vap}. At the training size of 2,000, ∼15% of the top ΔH_{vap} is identified for all acquisition functions other than random selection; the latter only identified ∼5% of formulations with the the top ΔH_{vap} values. Irrespective of expected improvement, greedy, or most uncertain acquisition function choice, we observe that formulation-property models can improve the identification of formulations with high ΔH_{vap} values 2-3 times faster than random selection.

Fig 5 demonstrated that ΔH_m was the most challenging to predict out of the three properties for formulation-property models. Fig. 7C shows that varying acquisition functions do not dramatically improve generalizability of formulation-property models to predict ΔH_m;

the most uncertain acquisition function achieved a highest test set R^2 of $\sim$0.80 when the training size is 2,000 examples, slightly higher than the random acquisition function. The greedy acquisition function struggled to create a generalized ΔH_m model and achieved a test set R^2 of $\sim$0.40 for all training sizes. Fig. 7F shows that the most uncertain acquisition function performed the best in identifying the formulations with the highest 5% of ΔH_m values, followed by expected improvement and greedy acquisition functions. Interestingly, the most uncertain acquisition function are not geared towards finding the maximum ΔH_m as compared to expected improvement and greedy acquisition functions, but the most uncertain acquisition function still outperformed the other two approaches by choosing candidates with the highest prediction uncertainty. The results in Fig. 7F show that even though the formulation-property models may not accurately predict ΔH_m at the small data scale, prediction uncertainties of ΔH_m could be useful to identify formulation candidates that are outside the domain of the training data and may have extrema of ΔH_m values. The most uncertain acquisition function achieves 2-3 times higher likelihood of selecting formulation candidates with high ΔH_m values as compared to random selection.

Fig. 7 demonstrates that formulation-property models are useful to identifying the next formulation candidates as compared to random selection irrespective of the acquisition function used. The selection of acquistion functions to use for an active learning workflow is highly dependent on the target property and how it is related to the underlying formulation structure. For simpler properties to predict with high test set R^2 close to 0.90, such as density or ΔH_{vap}, the greedy or expected improvement acquisition function generally perform well in identifying formulations with high property values. Conversely, for difficult to predict properties, such as ΔH_m, most uncertain and expected improvement acquisition functions that accounts for prediction uncertainty are better at identifying formulations that may be outside of the training domain and represent the extrema of property values. Overall, formulation-property relationships can serve as a powerful approach to rapidly screen formulations even with limited data, provide insight into important ingredient characteristics relevant to a target property through feature importance analysis, and provide suggestions of next best candidates in an active learning workflow to iteratively identify formulations satisfying a property criteria.

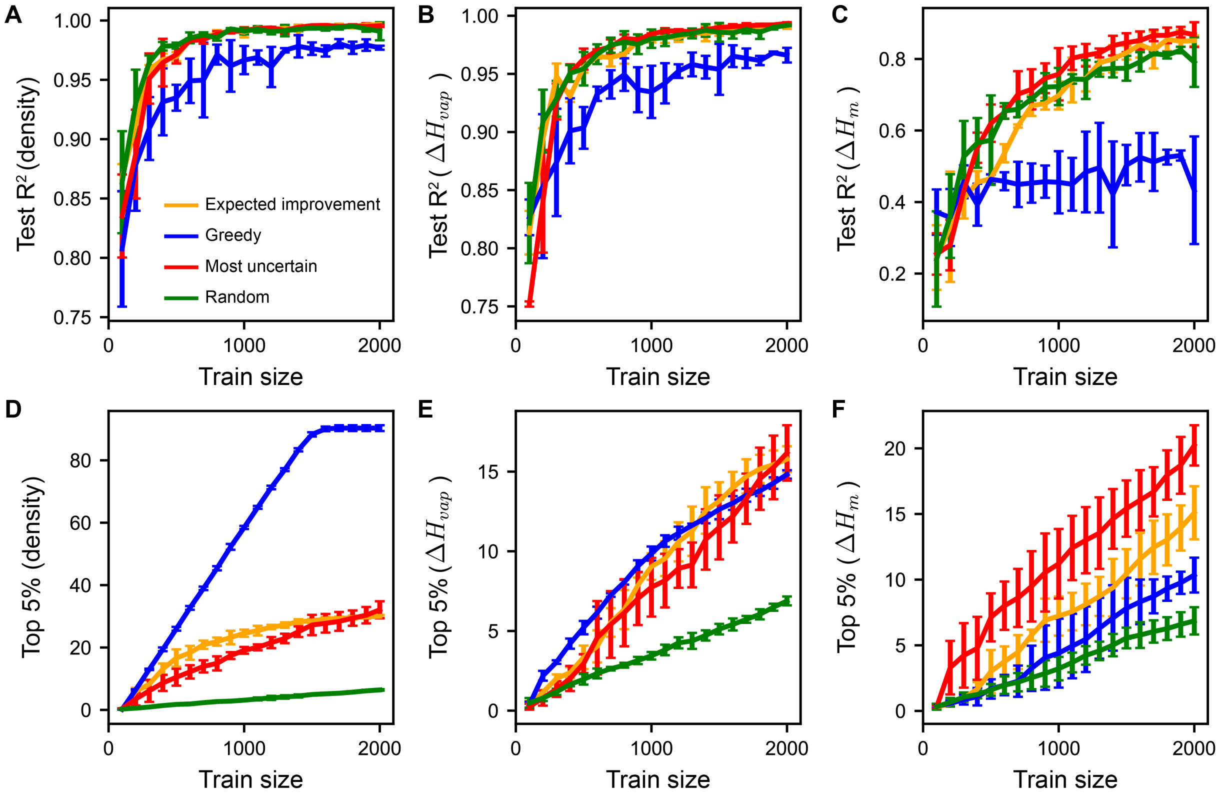


Figure 7: Active learning using formulation-property models. Left-out test set coefficient of determination (R^2) as a function of train size when training formulation-property models to maximize (**A**) density, (**B**) heat of vaporization (ΔH_{vap}), and (**C**) enthalpy of mixing (ΔH_m) using an active learning approach for expected improvement, greedy, most uncertain, and random acquisition functions. 10% of the 30,142 formulation examples were randomly selected as the left-out test set such that the test set contains unique formulations that are unseen in the training data pool. The percentage of formulations that are within the top 5% of the target property as a function of training size is shown for (**D**) density, (**E**) ΔH_{vap}, and (**F**) ΔH_m for the same acquisition functions used in A-C. The reported R^2 and top 5% is an average of three iterations of active learning runs with different random seeds, and the uncertainty of R^2 is the standard deviation of the different seeds.

4. Conclusion

In this work, we developed formulation-property relationships that input ingredient struture and composition to predict formulation properties, which is broadly applicable to a widerange of materials applications. First, we developed a formulation dataset by identifying miscible solvent mixtures based on miscibility tables and varied the number of components from pure to five component systems that results in a total of 30,142 formulation examples (Fig. 1 and Table 1). We developed three distinct formulation-property relationships, namely the formulation descriptor aggregation (FDA), formulation descriptor Set2Set (FDS2S), and the formulation graph (FG) approach (Fig. 2). Then, we performed high-throughput classical molecular dynamics (MD) simulations to generate formulation properties, such as density, heat of vaporization (ΔH_{vap}), and enthalpy of mixing (ΔH_m), all of which correlate with experimental data for specific solvent mixtures with a high correlation coefficient R^2 greater than ∼0.84 (Fig. 3). Using the large, simulation-derived formulation dataset, we found that increasing the number of components generally results in a narrower and denser property distribution, which suggests that mixtures of ingredients can allow for fine-tuning capabilities of the property space that is not possible with single component systems alone (Fig. 4). We benchmarked the different formulation-property approaches and found that the FDS2S approach performed the best in accurately predicting density, ΔH_{vap}, and ΔH_m at both small and large data scales, achieving a test set $R^2 \geq 0.96$ on all properties when trained with 90% of the data (Fig. 5). Analyzing the top features related to the formulation properties revealed that particular substructures were important, such as the inclusion of heavy halogen atoms to increase formulation density, inclusion of benzene, hydroxyl, or methylene groups to increase ΔH_{vap}, and inclusion of nitrogen-containing compounds to decrease ΔH_m (Fig. 6). Finally, when using formulation-property relationships in an active learning framework (Fig. 7), we observed that these models can rapidly identify the highest density, ΔH_{vap}, and ΔH_m values at least 2-3 times more likely than random guessing, which demonstrates that these formulation-property models are useful for designing formulations even when starting with a small dataset of less than a hundred examples.

The results highlight the strengths of both high-throughput MD simulations and machine learning approaches in identifying formulations with promising properties. MD simulations can rapidly compute formulation properties that accurately correlate with experiments, hence enabling a way to accurately generate formulation properties for a wide-range of material systems. These simulation-derived properties were useful to benchmark machine learning workflows to identify accurate formulation-property relationships. Aside from benchmarking purposes, these simulation-derived properties could be used as inputs into formulation-property relationships to predict more challenging formulation properties, such as viscosity of binary mixtures [14], charging efficiency in battery electrolytes [15, 16], fuel characteristics [8], or drug solubility in solvent mixtures [60]. Future work will focus on expanding the utility of these formulation-property relationships by encoding physics-based properties to improve model accuracy, enabling the optimization of formulations using the formulation-property relationships, and evaluating feature importance tools on graph-based formulation-property models.

Conflict of Interest

The authors declare no competing interests.

Data Availability

The formulation dataset is available upon request and will be available in the supporting information upon peer-review publication under the Creative Commons Non-Commercial 4.0 International (CC-BY-NC 4.0) Attribution License. This license allows for the use of the dataset and the creation of adaptations, exclusively for non-commercial purposes, provided that appropriate credit is given.

Supporting Information

The supporting information contains the comparison of formulation labels between molecular dynamics simulations and experiments, analysis of miscibility for binary mixtures using molecular dynamics simulations, best hyperparameters of formulation-property models when trained with 90% of the data, and description of the formulation dataset.

Acknowledgements

We are grateful to the data team at Schrödinger for their assistance in the data curation of the miscibility tables from the CRC handbook, namely Asela Chandrasinge, Mohammed Sulaiman, and Shawn Watts.

Author Contributions

A.K.C. and M.A.F.A. conceived the idea; A.K.C. performed the molecular dynamics simulations under guidance of M.A.F.A; A.K.C. wrote and generated figures for the manuscript; A.K.C., Z.K., E.M.C., and S.G. developed the code for the machine learning workflow; M.M. developed the code and graphical user interface within the Schrödinger suite; K.L. proposed the idea for using Set2Set models; all authors modified and approved the manuscript.

References

[1] Junko Habasaki. Atomistic molecular dynamics in polyethylene oxide and polymethyl methacrylate blends having significantly different glass transition temperatures. *International Journal of Applied Glass Science*, 13(3):347–358, 2022.

[2] Alex K Chew, Theodore W Walker, Zhizhang Shen, Benginur Demir, Liam Witteman, Jack Euclide, George W Huber, James A Dumesic, and Reid C Van Lehn. Effect of mixed-solvent environments on the selectivity of acid-catalyzed dehydration reactions. *ACS Catalysis*, 10(3):1679–1691, 2019.

[3] Theodore W Walker, Alex K Chew, Huixiang Li, Benginur Demir, Z Conrad Zhang, George W Huber, Reid C Van Lehn, and James A Dumesic. Universal kinetic solvent effects in acid-catalyzed reactions of biomass-derived oxygenates. *Energy & Environmental Science*, 11(3):617–628, 2018.

[4] Yaoyao Wei, Xueyu Wang, Lihua Dong, Guokui Liu, Qiying Xia, and Shiling Yuan. Molecular dynamics study on the effect of surfactant mixture on their packing states in mixed micelles. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, 631:127714, 2021.

[5] Chao Lu, Chuanjie Wu, Delaram Ghoreishi, Wei Chen, Lingle Wang, Wolfgang Damm, Gregory A Ross, Markus K Dahlgren, Ellery Russell, Christopher D Von Bargen, et al. Opls4: Improving force field accuracy on challenging regimes of chemical space. *Journal of chemical theory and computation*, 17(7):4291–4300, 2021.

[6] Mohammad Atif Faiz Afzal, Andrea R Browning, Alexander Goldberg, Mathew D Halls, Jacob L Gavartin, Tsuguo Morisato, Thomas F Hughes, David J Giesen, and Joseph E Goose. High-throughput molecular dynamics simulations and validation of thermophysical properties of polymers for various applications. *ACS Applied Polymer Materials*, 3 (2):620–630, 2020.

[7] Mohammad Atif Faiz Afzal, Aditya Sonpal, Mojtaba Haghighatlari, Andrew J Schultz, and Johannes Hachmann. A deep neural network model for packing density predictions and its application in the study of 1.5 million organic molecules. *Chemical science*, 10 (36):8374–8383, 2019.

[8] Nursulu Kuzhagaliyeva, Samuel Horváth, John Williams, Andre Nicolle, and S Mani Sarathy. Artificial intelligence-driven design of fuel mixtures. *Communications Chemistry*, 5(1):111, 2022.

[9] Dejun Jiang, Zhenxing Wu, Chang-Yu Hsieh, Guangyong Chen, Ben Liao, Zhe Wang, Chao Shen, Dongsheng Cao, Jian Wu, and Tingjun Hou. Could graph neural networks learn better molecular representation for drug discovery? a comparison study of descriptor-based and graph-based models. *Journal of cheminformatics*, 13(1):1–23, 2021.

[10] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

[11] Keith T Butler, Daniel W Davies, Hugh Cartwright, Olexandr Isayev, and Aron Walsh. Machine learning for molecular and materials science. *Nature*, 559(7715):547–555, 2018.

[12] Yuanbin Liu, Weixiang Hong, and Bingyang Cao. Machine learning for predicting thermodynamic properties of pure fluids and their mixtures. *Energy*, 188:116091, 2019.

[13] Emre Sevgen, Edward Kim, Brendan Folie, Ventura Rivera, Jason Koeller, Emily Rosenthal, Andrea Jacobs, and Julia Ling. Toward predictive chemical deformulation enabled by deep generative neural networks. *Industrial & Engineering Chemistry Research*, 60 (39):14176–14184, 2021.

[14] Camille Bilodeau, Andrei Kazakov, Sukrit Mukhopadhyay, Jillian Emerson, Tom Kalantar, Chris Muzny, and Klavs Jensen. Machine learning for predicting the viscosity of binary liquid mixtures. *Chemical Engineering Journal*, 464:142454, 2023.

[15] Vidushi Sharma, Maxwell Giammona, Dmitry Zubarev, Andy Tek, Khanh Nugyuen, Linda Sundberg, Daniele Congiu, and Young-Hye La. Formulation graphs for mapping structure-composition of battery electrolytes to device performance. *Journal of Chemical Information and Modeling*, 2023.

[16] Eduardo Soares, Vidushi Sharma, Emilio Vital Brazil, Young-Hye Na, and Renato Cerqueira. Capturing formulation design of battery electrolytes with chemical large language model. 2024.

[17] Kevin P Greenman, William H Green, and Rafael Gómez-Bombarelli. Multi-fidelity prediction of molecular optical peaks with deep learning. *Chemical science*, 13(4):1152–1162, 2022.

[18] Joonyoung F Joung, Minhi Han, Jinhyo Hwang, Minseok Jeong, Dong Hoon Choi, and Sungnam Park. Deep learning optical spectroscopy based on experimental database: potential applications to molecular design. *JACS Au*, 1(4):427–438, 2021.

[19] John R Rumble. Crc handbook of chemistry and physics, 103rd ed., 2022.

[20] Version 2023-2 Materials Science Suite. Schrödinger, llc: New york, 2022. URL https://www.schrodinger.com/platform/materials-science.

[21] Kevin J Bowers, Edmond Chow, Huafeng Xu, Ron O Dror, Michael P Eastwood, Brent A Gregersen, John L Klepeis, Istvan Kolossvary, Mark A Moraes, Federico D Sacerdoti, et al. Scalable algorithms for molecular dynamics simulations on commodity clusters. In *Proceedings of the 2006 ACM/IEEE Conference on Supercomputing*, page 84, 2006.

[22] Ryo Akasaka, Tomohiko Yamaguchi, and Takehiro Ito. Practical and direct expressions of the heat of vaporization for mixtures. *Chemical engineering science*, 60(16):4369–4376, 2005.

[23] Alex K Chew, Matthew Sender, Zachary Kaplan, Anand Chandrasekaran, Jackson Chief Elk, Andrea R Browning, H Shaun Kwak, Mathew D Halls, and Mohammad Atif Faiz Afzal. Advancing material property prediction: Using physics-informed machine learning models for viscosity. 2024.

[24] Lei Qun-Fang, Hou Yu-Chun, and Lin Rui-Sen. Correlation of viscosities of pure liquids in a wide temperature range. *Fluid Phase Equilibria*, 140(1-2):221–231, 1997.

[25] Jianxing Dai, Xiaofeng Li, Lifeng Zhao, and Huai Sun. Enthalpies of mixing predicted using molecular dynamics simulations and opls force field. *Fluid Phase Equilibria*, 289 (2):156–165, 2010.

[26] Sonia M Aguilera-Segura, Francesco Di Renzo, and Tzonka Mineva. Structures, intermolecular interactions, and chemical hardness of binary water–organic solvents: a molecular dynamics study. *Journal of molecular modeling*, 24:1–14, 2018.

[27] Ying Yang, Kun Yao, Matthew P Repasky, Karl Leswing, Robert Abel, Brian K Shoichet, and Steven V Jerome. Efficient exploration of chemical space with docking and deep learning. *Journal of Chemical Theory and Computation*, 17(11):7106–7119, 2021.

[28] Benchmark study of deepautoqsar, chemprop, and deeppurpose on the admet subset of the therapeutic data commons. https://www.schrodinger.com/sites/default/files/22_086_machine_learning_white_paper_r4-1.pdf, 2022. Accessed: 2024-05-04.

[29] Logan Ward, Alexander Dunn, Alireza Faghaninia, Nils ER Zimmermann, Saurabh Bajaj, Qi Wang, Joseph Montoya, Jiming Chen, Kyle Bystrom, Maxwell Dylla, et al. Matminer: An open source toolkit for materials data mining. *Computational Materials Science*, 152:60–69, 2018.

[30] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015.

[31] Mingjian Wen, Samuel M Blau, Evan Walter Clark Spotte-Smith, Shyam Dwaraknath, and Kristin A Persson. Bondnet: a graph neural network for the prediction of bond dissociation energies for charged molecules. *Chemical science*, 12(5):1858–1868, 2021.

[32] Rishabh Debraj Guha, Santiago Vargas, Evan Walter Clark Spotte-Smith, Alexander R Epstein, Maxwell Christopher Venetos, Mingjian Wen, Ryan Kingsbury, Samuel M Blau, and Kristin Persson. Hepom: A predictive framework for accelerated hydrolysis energy predictions of organic molecules. In *AI for Accelerated Materials Design-NeurIPS 2023 Workshop*, 2023.

[33] CL Mellor, RL Marchese Robinson, Romualdo Benigni, David Ebbrell, SJ Enoch, JW Firman, JC Madden, Gopal Pawar, Chihae Yang, and MTD Cronin. Molecular fingerprint-derived similarity measures for toxicological read-across: Recommendations for optimal use. *Regulatory Toxicology and Pharmacology*, 101:121–134, 2019.

[34] Greg Landrum et al. Rdkit. *Q2. https://www. rdkit. org*, 2010.

[35] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939785. URL http://doi.acm.org/10.1145/2939672.2939785.

[36] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[37] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems*, 28, 2015.

[38] Boris Knyazev, Graham W Taylor, and Mohamed Amer. Understanding attention and generalization in graph neural networks. *Advances in neural information processing systems*, 32, 2019.

[39] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[40] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[41] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. Self-attention graph pooling. In *International conference on machine learning*, pages 3734–3743. PMLR, 2019.

[42] Frederik Diehl. Edge contraction pooling for graph neural networks. *arXiv preprint arXiv:1905.10990*, 2019.

[43] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[44] 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 E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[45] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[46] Version 2024-2 Materials Science Suite. Schrödinger, llc: New york, 2024. URL https://www.schrodinger.com/platform/materials-science.

[47] Andrew F Zahrt, Jeremy J Henle, and Scott E Denmark. Cautionary guidelines for machine learning studies with combinatorial datasets. *ACS Combinatorial Science*, 22 (11):586–591, 2020.

[48] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

[49] Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2(1):2522–5839, 2020.

[50] Raquel Rodríguez-Pérez and Jürgen Bajorath. Interpretation of compound activity predictions from complex machine learning models using local approximations and shapley values. *Journal of medicinal chemistry*, 63(16):8761–8777, 2019.

[51] Pauric Bannigan, Zeqing Bao, Riley J Hickman, Matteo Aldeghi, Florian Häse, Alán Aspuru-Guzik, and Christine Allen. Machine learning models to accelerate the design of polymeric long-acting injectables. *Nature Communications*, 14(1):35, 2023.

[52] Derek van Tilborg and Francesca Grisoni. Traversing chemical space with active deep learning. 2023.

[53] Hadi Abroshan, H Shaun Kwak, Anand Chandrasekaran, Alex K Chew, Alexandr Fonari, and Mathew D Halls. High-throughput screening of hole transport materials for quantum dot light-emitting diodes. *Chemistry of Materials*, 35(13):5059–5070, 2023.

[54] Zhonglin Cao, Simone Sciabola, and Ye Wang. Large-scale pretraining improves sample efficiency of active learning-based virtual screening. *Journal of Chemical Information and Modeling*, 2024.

[55] Kalju Kahn and Thomas C Bruice. Parameterization of opls–aa force field for the conformational analysis of macrocyclic polyketides. *Journal of computational chemistry*, 23(10):977–996, 2002.

[56] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

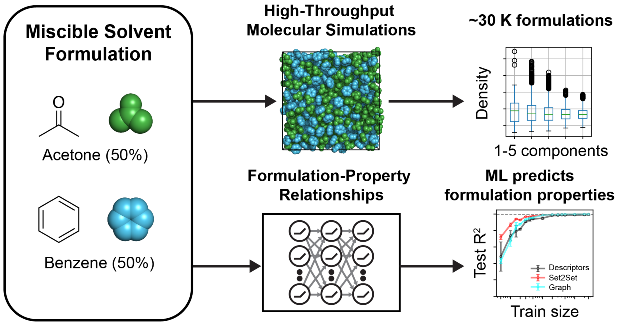
[57] Alex K Chew, Joel A Pedersen, and Reid C Van Lehn. Predicting the physicochemical properties and biological activities of monolayer-protected gold nanoparticles using simulation-derived descriptors. *ACS nano*, 16(4):6282–6292, 2022.

[58] Tony P Tauer and C David Sherrill. Beyond the benzene dimer: an investigation of the additivity of π- π interactions. *The Journal of Physical Chemistry A*, 109(46): 10475–10478, 2005.

[59] Alex K Chew and Reid C Van Lehn. Effect of core morphology on the structural asymmetry of alkanethiol monolayer-protected gold nanoparticles. *The Journal of Physical Chemistry C*, 122(45):26288–26297, 2018.

[60] Zeqing Bao, Gary Tom, Austin Cheng, Alán Aspuru-Guzik, and Christine Allen. Towards the prediction of drug solubility in binary solvent mixtures at various temperatures using machine learning. 2024.

Table of Contents Graphic


Miscible Solvent Formulation
O
Acetone (50%)
Benzene (50%)
High-Throughput Molecular Simulations
~30 K formulations
Density
1-5 components
Formulation-Property Relationships
ML predicts formulation properties
Test R²
Descriptors
Set2Set
Graph
Train size