Machine Learning-Guided Space-filling Designs for High Throughput Liquid Formulations Development

Aniket Chitre,[1,2,3] David C. Woods,[4] Alexei A. Lapkin[1,2*]

[1] *Department of Chemical Engineering and Biotechnology, University of Cambridge, Philippa Fawcett Drive, Cambridge CB3 0AS, United Kingdom*

[2] *Cambridge Centre for Advanced Research and Education in Singapore, CARES Ltd. 1 CREATE Way, CREATE Tower #05-05, Singapore 138602, Singapore*

[3] *Institute of Materials Research and Engineering, Agency for Science, Technology and Research (A∗STAR), Singapore 138634, Singapore*

[4] *Southampton Statistical Sciences Research Institute and School of Mathematical Sciences, University of Southampton, Southampton SO17 1BJ, UK*

*Corresponding author email address: aal35@cam.ac.uk

Abstract

Liquid formulations design typically involves searching a high-dimensional space, owing to the combinatorial selection of ingredients from a larger subset of available ingredients, with a relatively limited experimental budget. Therefore, we need to efficiently select the most informative experiments. These experiments need to optimise the composition of these industrially-manufactured products towards customer defined target-properties. Consequently, we have a mixed discrete-continuous Design of Experiments (DoE) problem, for which there are few computationally efficient solutions, with the exception of maximum projection designs with quantitative and qualitative factors (MaxProQQ). However, such purely space-filling designs can select experiments in infeasible regions of the design space. Here, we explore a system of shampoo formulations, where only stable products are considered feasible. We show a weighted-space filling design, where predictive phase stability classifiers are trained for difficult-to-formulate sub-systems, to guide these experiments to regions of feasibility, whilst simultaneously optimising for chemical diversity through building on MaxProQQ.

Keywords: *Design of experiments; machine learning; liquid formulations; phase stability*

Introduction

Liquid formulations are complex multi-component mixtures where the ingredients have been selected, processed, and combined in a specific way to obtain well-defined target properties, functionality and performance (Conte et al., 2011). Selecting which ingredients to use and what concentrations to use them at constitutes a mixed discrete-continuous design problem. Typically, these products, which are produced across several industries (*e.g.*, consumer care, agrochemical, pharmaceutical; Bagajewicz et al., 2011; Bernardo and Saraiva, 2005; Gani, 2004; Narayanan et al., 2021; Taifouris et al., 2020), are developed through trial-and-error by specialists with extensive experience in the given domain. Industry seeks a more systematic methodology to develop formulated products, particularly, as we wish to formulate novel products, either for enhanced performance and functionality (Gani and Ng, 2015; Martín and Martínez, 2013), or for environmental reasons (Jessop et al., 2015; Kelly, 2023). We aim to train predictive surrogate models for liquid formulations design, so in simple terms, we needed to generate the most informative set of experimental data which represents our design space.

As formulations design is a combinatorial problem of ingredients selection, we often have a very large design space to explore, yet we have a limited experimental budget. This is generally true for chemical/chemical engineering problems as experiments are time-, resource-, and labour-intensive, but particularly for formulations design, as developing a fully automated, high-throughput liquid formulations workflow is very challenging (Cao et al., 2021). Therefore, we needed an efficient design of experiments (DoE) methodology.

Here, we prepared shampoo formulations with two surfactants, a conditioning polymer, and a thickener in a base of water, as shown in Figure 1. This chemical system is similar to a previous study from our group (Cao et al., 2021), but with an extended set of ingredients to choose from: 12 surfactants, four conditioning polymers, and two thickeners, *i.e.*, 528 possible ingredient combinations, which necessitated the development of new methods. We have separately detailed our high-throughput liquid formulations workflow to prepare and characterise these formulations (Chitre et al., 2024). Our overall goal is to develop accurate property (phase stability, turbidity, rheology) prediction models across the entire design space; therefore, our DoE objective was to develop an optimally space-filling design.

In an earlier study, we developed a "Bridge DoE" for liquid formulations, which looked at selecting a ternary combination of surfactants from a set of five, *i.e.*, 10 possible combinations (Cao et al., 2023). This prior work demonstrated balanced exploration and exploitation of the design space, however, relied on an expensive objective function (Eqn. 1) which scales particularly poorly to an over 50-fold higher dimensionality design space in this study.

$$\emptyset(\boldsymbol{D}) = \omega \log\left(\tilde{E}[\mathrm{U}(\boldsymbol{D})]\right) + (1-\omega)\log\left(\mathrm{d}(\boldsymbol{D})\right) \quad (1)$$

Here, $\omega \in [0,1]$ is a weighting between the two parts of the bridge DoE objective function relating to the phase stability and viscosity test results, respectively; $\boldsymbol{D}$ is the design scaled to be between 0 and 1; $\tilde{E}[\mathrm{U}(\boldsymbol{D})]$ is the expected utility relating to estimation of, or prediction from, a machine learning model of the phase stability response and $\mathrm{d}(\boldsymbol{D})$ is the average Euclidean distance between all possible pairs of rows in $\boldsymbol{D}$. Firstly, all possible pairs of rows in $\boldsymbol{D}$ combinatorially explodes with the set of ingredients to choose from and secondly, the expected utility function was approximated by Monte Carlo integration which was expensive to evaluate and becomes more problematic as the design space grows. Therefore, the Bridge DoE was intractable for this work.

We therefore returned to traditional space-filling designs (Johnson et al., 1990; McKay et al., 1979), for which Joseph (2016) provides an excellent review. In practice, Maximin Latin Hypercube Designs (Mm LHD) are the most commonly used due to their simplicity and availability in software packages (Morris and Mitchell, 1993). However, a LHD only works with continuous factors. Therefore, sliced Latin hypercube designs (SLHD) were introduced (Qian, 2012), which are a type of LHD that can be further partitioned into t smaller LHDs called slices where t is the number of all possible combinations of the categorial factors. The method was improved to find an alternative, more computationally efficient construction of the SLHD (Ba et al., 2015), yet as the number of nominal factors increases, t increases exponentially, and so this method is limited for its application to formulations design. Furthermore, whilst maximin (S)LHDs have optimal space-filling properties in the full p-dimensions of a design problem and uniform 1-d projections, their space filling properties in lower-dimensional projections, $2, \ldots, p-1$, can be poor. In the formulations context, this could mean you have good space-filling properties when you consider the full design space, including all ingredients, but for example, if you wanted to fix one of the ingredients, say a new bio-sourced thickener molecule, and investigate the response of the surfactants compatible

with this thickener, then in this reduced dimensional space, (S)LHDs would not ensure good space-filling properties and you could have several experiments correlated with eachother, *i.e.*, not providing new information. Consequently, an alternative design criterion, Maximum Projection (MaxPro), with equal computational cost to the Morris & Mitchell Mm LHD criterion, was proposed, which ensures good projections to all the subspaces of the factors (Joseph et al., 2015). This method was further extended as MaxProQQ to work with both quantitative and qualitative factors (Joseph et al., 2020), as required for formulations design.

MaxProQQ could be directly used to generate space-filling liquid formulation designs, however, purely space-filling methods can result in the selection of points in areas of little relevance, for example, where it is known no response can occur (Bowman and Woods, 2013). For this work, we are interested in measuring the phase stability, turbidity, and rheology of the prepared formulations, however, only the stable formulations are characterised for their turbidity and rheology, as these measurements are not meaningful for inhomogeneous mixtures (Chitre et al., 2024). This is summarised in Figure 1 which shows that a set of experiments are executed according to the DoE. These formulations are then assessed for their phase stability, with the stable samples being titrated to an industrially specified pH target (Chitre et al., 2023). We then waited for a fixed amount of time (pre-agreed with our industrial partner as 36 hours) before re-assessing the formulations' stabilities and characterising the stable products for their turbidity and rheology. So, every unstable formulation resulted in no turbidity and rheology data, without which we cannot train any property prediction models for these targets. Therefore, we used and present a weighted-space filling design to guide our experiments to regions of phase stability. However, we do not know *a priori* which regions of the formulations design space will be stable. Therefore, we used an active learning approach to train a predictive phase stability classifier across the design space, which would be used to guide difficult-to-formulate sub-systems to regions of stability, as part of a machine learning-guided DoE (ML-guided DoE). Here the DoE component of the method was developed using the MaxProQQ package. With this study, we show that we are able to optimise for the chemical diversity and phase stability in our formulations dataset.

The remainder of the paper is structured as follows. Firstly, we present the methodology for our phase-stability guided MaxProQQ designs, including details for featurising our liquid formulations and training predictive stability classifiers. We then show in the Results and Discussion section, that we have been able to optimise our systems towards phase stability and

demonstrate the spread and coverage of our designs. Finally, we present the performance of ML stability classifiers and discuss the chemical interpretability of our results.

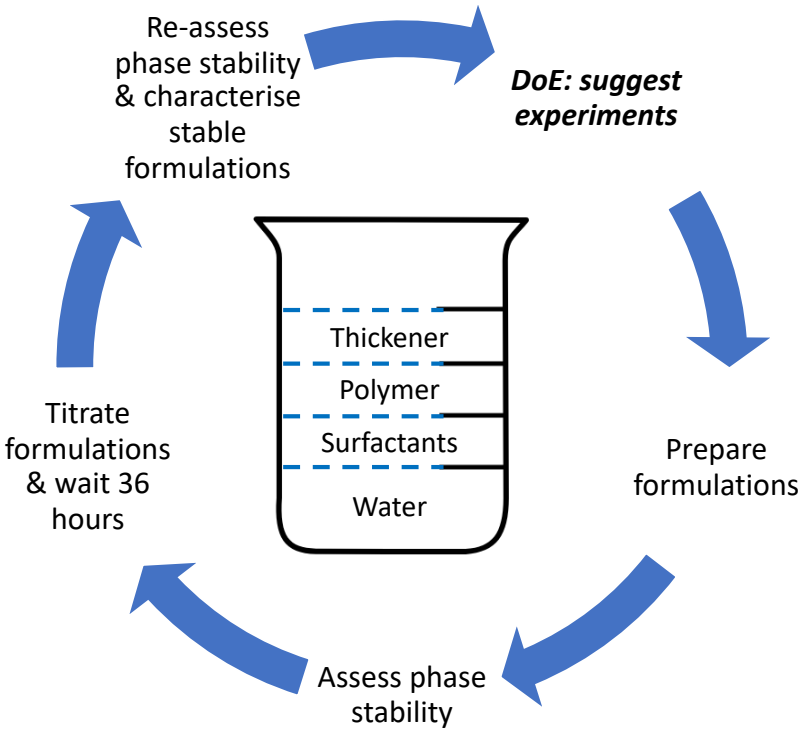

Figure 1. Overview of liquid formulations workflow driven by an ML-guided DoE method for a (weighted-) space filling design towards phase stability and chemical diversity.

Materials and Methods

Formulations Featurisation

We used commercial formulation ingredients as received from BASF. These materials with their chemical structures are fully detailed in a separate work, as part of the Supplementary Information (SI) of our formulations workflow and dataset (Chitre et al., 2024). We formulated from a set of 12 surfactants, four conditioning polymers (P_1 = Luviquat® Excellence, P_2 = Dehyquart® CC6, P_3 = Dehyquart® CC7 Benz, P_4 = Salcare® Super 7), and two thickeners (T_1 = Arlypon® TT, T_2 = Arlypon® F). Of these, we note that P_1 and P_2 were relatively highly charged cationic polyelectrolytes, whilst P_3 and P_4 had a lower charge density. This will be used later.

In order to develop a phase stability classifier for a weighted-space filling design, as introduced earlier, we needed to featurise our formulations in a machine-readable manner (Wigh et al.,

2022). Unfortunately, effectively featurising macromolecules, such as the polymeric and thickener ingredients, is an open question with many promising recent studies (Kim et al., 2018; Kuenneth and Ramprasad, 2023; Lin et al., 2019) but no general solution to date. Mixtures of such molecules – *i.e.*, formulations, are even more difficult to represent. Currently, the simplest approach is to directly take the concentrations of polymer and thickener added, and a one-hot encoding of the ingredients, as in our previous work (Cao et al., 2021). Since we are presently unable to find a chemically suitable featurisation for the polymer and thickener ingredients, and to reduce the dimensionality of the DoE problem, we split the design space into eight distinct sub-systems for each possible polymer, thickener combination: $(P_1, T_1), (P_1, T_2), (P_2, T_1) \dots (P_4, T_2)$. Intuitively, we could also expect fixed combinations of polymer and thickener to exhibit some chemically similar behaviours with the different classes of surfactant molecules (anionic/non-ionic/amphoteric/cationic). This step of fixing the polymer and thickener reduced our design problem by two dimensions to a 5D problem: i) four continuous variables for the concentrations of the surfactants, polymer, and thickener $(C_{S1}, C_{S2}, C_P, C_T)$; and ii) one discrete variable with 66 levels representing the choice of surfactant pair.

One of the limitations of our previous work, where we have featurised formulations without any structural information, is that the trained models cannot be generalised to any new ingredients without effectively re-starting the experimental campaign (Cao et al., 2021). Hence, we would like to find a more generalisable featurisation for the surfactants, so that we can make in-silico predictions for a new drop-in replacement. As small organic molecules, there are many methods for featurising this type of ingredient with i) string-based representations, *e.g.*, SMILES (Öztürk et al., 2016; Schwartz et al., 2013; Vidal et al., 2005); ii) molecular graphs (Qin et al., 2021; Yang et al., 2019); and iii) molecular features (from 0D to 3D descriptors) (Abooali and Soleimani, 2023; Consonni and Todeschini, 2010; Ghiringhelli et al., 2015; Seddon et al., 2022). This list is not exhaustive. There are in particular many different cheminformatics packages (Bray et al., 2020; Moriwaki et al., 2018; O'Boyle et al., 2011; Yap, 2011) that can enumerate large numbers of descriptors, which should be feature engineered down to a more sensible subset relative to the dataset size available for training. However, many of these featurisations require large training datasets or are not directly interpretable. With formulations design, we are constrained to generating hundreds, not thousands, of samples, even with state-of-the-art lab facilities, and so we developed a more chemically meaningful featurisation based on the surfactant functional groups, as shown in Figure 2. This

was hypothesised to improve model performance and explainability, as illustrated in the Results & Discussion section.

We used our physical chemistry knowledge that a surfactant's behaviour is primarily governed by its head group and chain length (Kronberg et al., 2014). As highlighted in Figure 2a for an example surfactant (Texapon® SB 3 KC), we can enumerate all the distinct functional groups (FG) – either algorithmically (Ertl, 2017), or by hand since we only have twelve surfactants here – and count their frequency. This data is summarised in the SI of our formulations dataset (Chitre et al., 2024). Additionally, not highlighted in Figure 2a, is the length of the alkyl chain, $(CH_2)_x$. As we worked with real, industrial materials, these surfactant ingredients have a distribution of chain lengths which we characterised via UPLC-MS, as detailed in the SI of our previous work (Chitre et al., 2024). This FG and chain length data is represented by the orange matrix in the top-right hand corner of Figure 2b. This shows that we can take the matrix dot product of the experimental surfactant concentrations and FG data to re-express the surfactants in terms of concentrations of functional groups and append the polymer and thickener concentrations to generate a suitable featurisation for a sub-system of formulation samples.

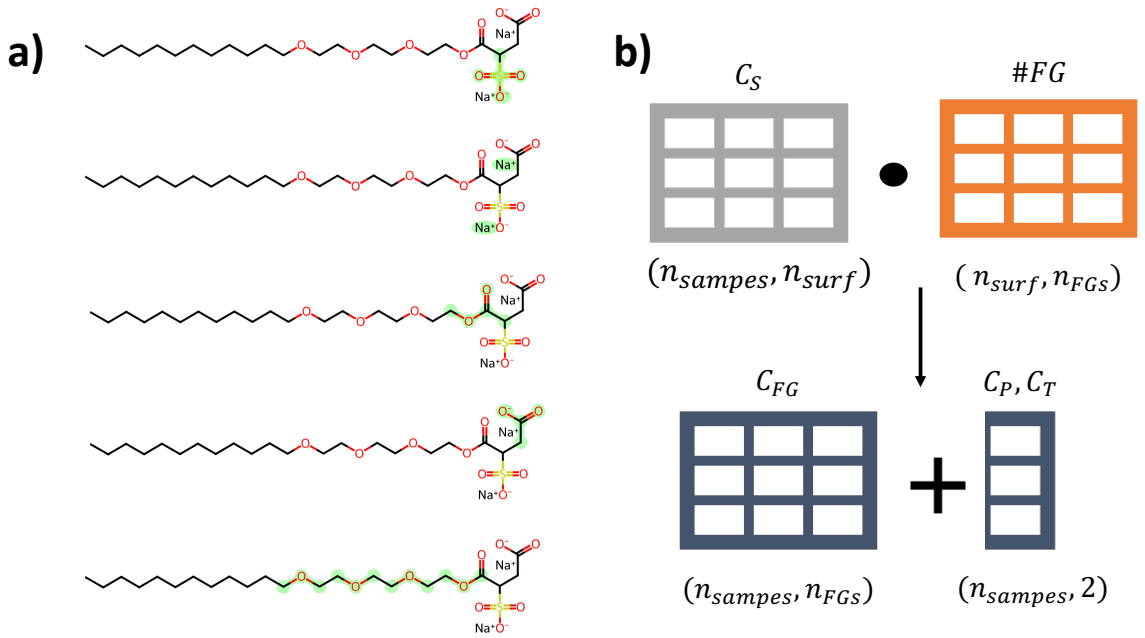

Figure 2. Surfactants featurisation via functional groups (FG) (a) Counting the unique FGs in an example surfactant ingredient; and (b) showing how this is used with the experimental data to encode the formulations by their surfactant FG, polymer, and thickener concentrations. The (m, n) below the dataframes show dimensions of m rows and n columns.

Phase Stability-guided MaxProQQ Designs

(1) Generate a random starting design $\mathbf{D_{init}}$ and candidate set $\mathbf{C}$ of n_{init} and N points, respectively, where $N = 360{,}000$.

- Use MaxPro's `CandPoints` with 4 continuous factors and 1 nominal factor with 66 levels ($^{12}C_2$ surfactant pair combinations).

(2) Convert $\mathbf{D_{init}}$ which has 4 continuous factors $\in [0, 1]$ and a nominal factor, which is a one-hot encoding of the surfactants pair, into $\mathbf{D^*_{init}}$, the initial experimental design which is a CSV file readable by our Opentrons OT-2 protocol.

- Convert the design variables based on the desired concentration (conc.) bounds for the surfactants (S), polymer (P), and thickener (T) ingredients:
  i. $8 \leq S_1, S_2 \leq 13 \frac{w}{w}\%$
  ii. $1 \leq P \leq 3 \frac{w}{w}\%$
  iii. $1 \leq T \leq 5 \frac{w}{w}\%$

- Use a look-up table for the surfactant pair encoding.

(3) Perform the experiments and record their compositions. Use the inverse operations of step 2 to generate the prepared design $\mathbf{D}$.

(4) IF $< \chi$ of the initial formulations are stable:

- Train a phase-stability classifier on the experimental dataset with the:
  i. Surfactant conc. converted to functional group (FG) conc.
  ii. Polymer and thickener concentrations used as is, with both (i) and (ii) pre-processed with min-max scaling.
  iii. Phase stability used as the output y to train the classifier.

- Test a variety of machine learning (ML) models and select the best performing one to make *in silico* predictions for the stability of samples in the candidate set $\mathbf{C}$.

- Filter $\mathbf{C}$ by a phase stability criterion to a restricted candidate set $\mathbf{C}^*$which is a subset of samples with a higher probability of stability. Use this for step 5.

ELSE:

PASS and use $\mathbf{C}$ for step 5.

(5) Use `MaxProAugment` to suggest n_{add} additional experiments ($\mathbf{D}^*$) using a one-at-a-time greedy optimisation procedure based on the existing dataset $\mathbf{D}$ and (restricted) candidate design. Go to step 3 and repeat until sufficient samples are prepared. Then move onto the next polymer-thickener sub-system.

Scheme 1. ML-guided DoE algorithm for a particular (polymer, thickener) sub-system.

Scheme 1 outlines the algorithm for the ML-guided DoE method which was motivated in the Introduction. We firstly randomly generate 360,000 points in the design space (using MaxPro's `CandPoints` function); these points are taken to represent the total set of possible experiments. We used a fixed random seed to always generate the same candidate set, **C**, across all iterations of the active learning loop for a particular polymer, thickener sub-system, and also for all the sub-systems investigated. Therefore, we are always proposing experiments from the same potential design space. Our industrial partner had recommended a set of concentration bounds for us to formulate within, as detailed under step (2) in Scheme 1. If we discretised these bounds in 0.5 w/w% intervals and accounted for the surfactants selection problem, then we have approximately 360,000 combinations; hence, the size selected for **C**. Here, 0.5 w/w% was determined to be an appropriate step which could be comfortably resolved by our experimental procedure - automated viscous liquids handling on the Opentrons OT-2 robot (Chitre et al., 2024). We note, we could accurately determine the composition of the prepared formulations; however, we could not always accurately dispense the target amounts specified from our DoE, especially for the highly viscous formulation ingredients. Therefore, on each iteration of the DoE, we suggested the next batch of experiments based on the actual, recorded compositions.

We start with an initial design for a fixed sub-system of polymer and thickener, $\mathbf{D_{init}}$, with n_{init} number of points, where n_{init} was typically set to 36 samples, the maximum throughput of our formulations workflow in a week. We analysed the proportion of stable formulations in our initial design. If $< \chi$ of the formulations were stable, these sub-systems were defined as "difficult-to-formulate", in which case we applied a phase-stability guided DoE strategy. This ensured a large number of experiments would not be wasted without generating any turbidity or rheology data. Otherwise, we preferred a purely space-filling design for the other sub-systems, as this imposes no restriction, allowing better modelling of the entire design space. For this study, χ = 40%.

For the difficult-to-formulate sub-systems, we would train a predictive phase stability classifier, using a featurisation of the experimental data as explained in the previous sub-section. Details for how we trained this classifier are described in the SI. We would then use this classifier to predict the phase stability of each point in our candidate set, **C**, and apply a phase stability cut-off (between 0 and 1) to drop any points without a minimum probability of stability, to generate a restricted candidate set, $\mathbf{C}^*$. This phase-stability cut-off would be

modified on each iteration of the DoE as explained in the Results & Discussion section. For the other, more stable sub-systems, we kept the original candidate set. Finally, we used a greedy search algorithm to select the next batch of experiments out of the (restricted) candidate design to best augment the already collected experimental data based on the MaxPro criterion (using the `MaxProAugment` function). If 36 additional experiments are requested, instead of performing an expensive optimisation to simultaneously calculate the 36 best points, a one-at-a-time greedy optimisation procedure sequentially suggests points 1 to 36. There is a small trade-off in optimality for substantially increased computational efficiency, as we wish to be able to generate designs on-the-fly in a high-dimensional design space for high-throughput experimental campaigns. The code for the complete method is provided in the SI.

Results and Discussion

Formulating Stable Systems

The method outlined above was for a particular polymer, thickener sub-system. This was applied on all eight sub-systems; however, for the formulations with P_2 (Dehyquart® CC6) too few formulations were stable, so we could not train an accurate stability classifier. We prepared 174 samples over two months with this polymer, but less than 15% of these formulations were stable. There are some domains, *e.g.*, in finance detecting credit card fraud, where we can work with a heavily imbalanced dataset, for *e.g.*, < 1% of the data may be a fraudulent transaction, yet we can develop a predictive ML model (Bin Sulaiman et al., 2022; Tran and Dang, 2021). However, this is a big data problem, a luxury we are not afforded when working with formulations, therefore, this highlights the continued importance of the formulator's expertise to suggest suitable concentration bounds for us to explore within. Henceforth, we exclude sub-systems (P_2, T_1) and (P_2, T_2) from our results for difficult-to-formulate sub-systems, as we could not apply the weighted-space filling design without a predictive stability classifier. We, therefore, had three sub-systems for which our ML-guided DoE was used, as shown in Figure 3. For the rest, a purely space-filling MaxProQQ design was used throughout. We highlight, with the exception of (P_4, T_1), the difficult-to-formulate sub-systems were primarily those prepared with the highly charged cationic polyelectrolytes, P_1 and P_2 because these would often form coacervates with the anionic surfactants in our set of ingredients.

Figure 3 shows that with our phase stability-guided MaxProQQ designs, we could tune our experiments to stable regions of the design space across all three sub-systems. The proportion

of stable formulations in a batch are coloured with a hue to represent the phase stability cut-off. This threshold was progressively increased over each round. Initially, we want a low threshold to favour exploration of the design space, and as we have more experimental data and train a better stability classifier, we can exploit this model to strongly bias our formulations to regions of stability, as seen if you compare the first and last points across all three sub-systems. Note, for the first point, the stable in round and overall stable (%) are equivalent and the initial design was generated by random sampling.

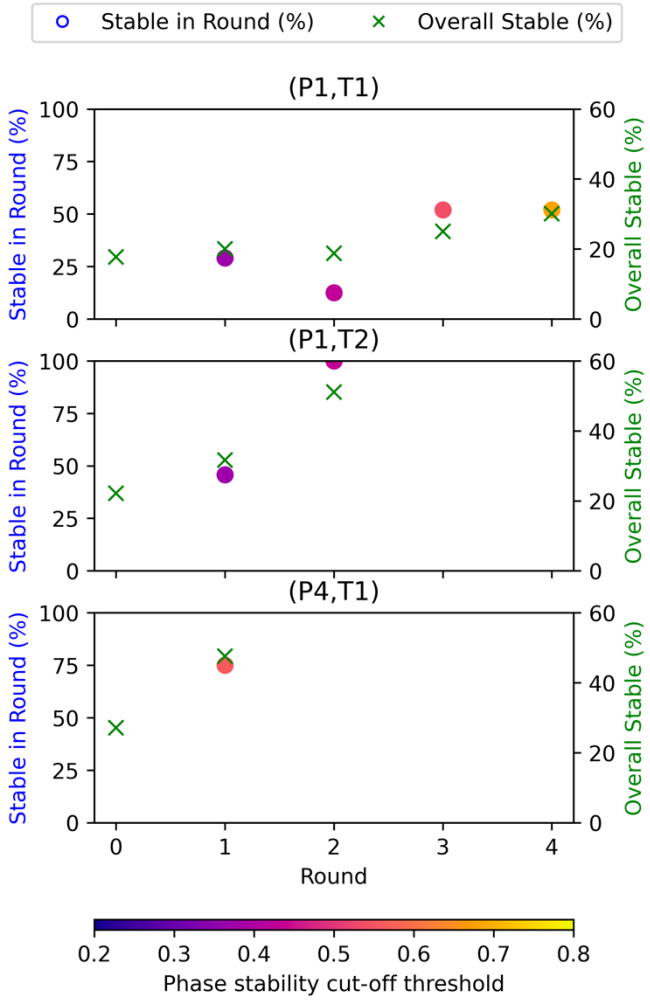

Figure 3. ML-guided DoE used to bias formulation sub-systems towards regions of phase stability.

The phase stability cut-off is used to tune the balance between complete exploration of the design space (low threshold) and exploitation of the stability classifier to predicted stable regions in the design space (high threshold). This stability cut-off is increased on each iteration.

The only example from Figure 3 where our ML-guided DoE fails to increase the stable in round (%) is after the first round of experiments with (P_1, T_1). We go from 29 to 13% of formulations being stable in round 1 *vs.* 2. However, this can be clearly explained as shown by Table S1 in the SI. Tables S1 - S3 provide the full set of phase stability-guided DoE results for all three sub-systems, complementary to Figure 3. Initially, we chose to apply a cut-off as some top x% of experiments. On round one of (P_1, T_1), which was the first sub-system we explored, we restricted the candidate set to the top 20% of stable predicted experiments; however, this cut-off was equivalent to a 0.29 phase stability threshold, which would still include a majority of unstable formulations, as seen in round two. We, therefore, soon switched to only defining the phase stability cut-off as a predicted probability of stability between zero and one, so we could more clearly control the degree of stability tuning. Additionally, as seen in Table S1, the best classifier at round one had ROC AUC and F_1 scores of 0.62 and 0.73, respectively, which corresponds to a moderately predictive classifier. As highlighted above, it is essential to be able to develop a highly predictive classifier, otherwise, we cannot apply this weighted search strategy effectively. By contrast, the initial classifiers trained for (P_1, T_2) and (P_4, T_1) are excellent (see Tables S2 and S3), and so we could successfully guide our difficult-to-formulate sub-systems to regions of stability in just one or two iterations.

Design Coverage and Spread

The other objective of our ML-guided DoE was to optimally space-fill so that we can develop predictive surrogate models over the entire formulations design space. We already established that we fixed the polymer and thickener for a particular sub-system and explored all these sub-systems, therefore, we look at the spread of surfactants used in Figure 4. We prepared a total of 384 formulations for the three difficult-to-formulate sub-systems, as identified earlier, and 438 further samples for the remaining five sub-systems. The dashed lines in Figure 4 show the expected number of samples per surfactant if we had uniformly sampled the ingredients. We observe that for the purely space-filling designs, our surfactants' distribution is very close to this expected value, showing excellent space-filling properties. By comparison, and as we would expect, we have a non-uniform distribution for the stability-guided experiments as our classifier learned that certain surfactant(s) would lead to unstable results with a particular polymer, thickener, or indeed, another surfactant. For example, as stated earlier P_1 and P_2 are highly charged cationic polyelectrolytes, so Texapon® SB 3 KC, Plantapon® ACG 50, and Plantapon® LC7, which are anionic surfactants would often form coacervates with these

ingredients, and therefore you see they are under sampled for the stability-guided designs. And following this argument, Dehyquart® A-CA, the only cationic surfactant in the set, was particularly favoured for the stability-guided experiments. Despite this, we have still sampled all the ingredients relatively well, which was achieved through modifying the phase stability cut-off to balance exploration of our design space, which leads to a more space-filling design, and simultaneously exploiting our classifier to drive the experiments to feasible regions of the design space.

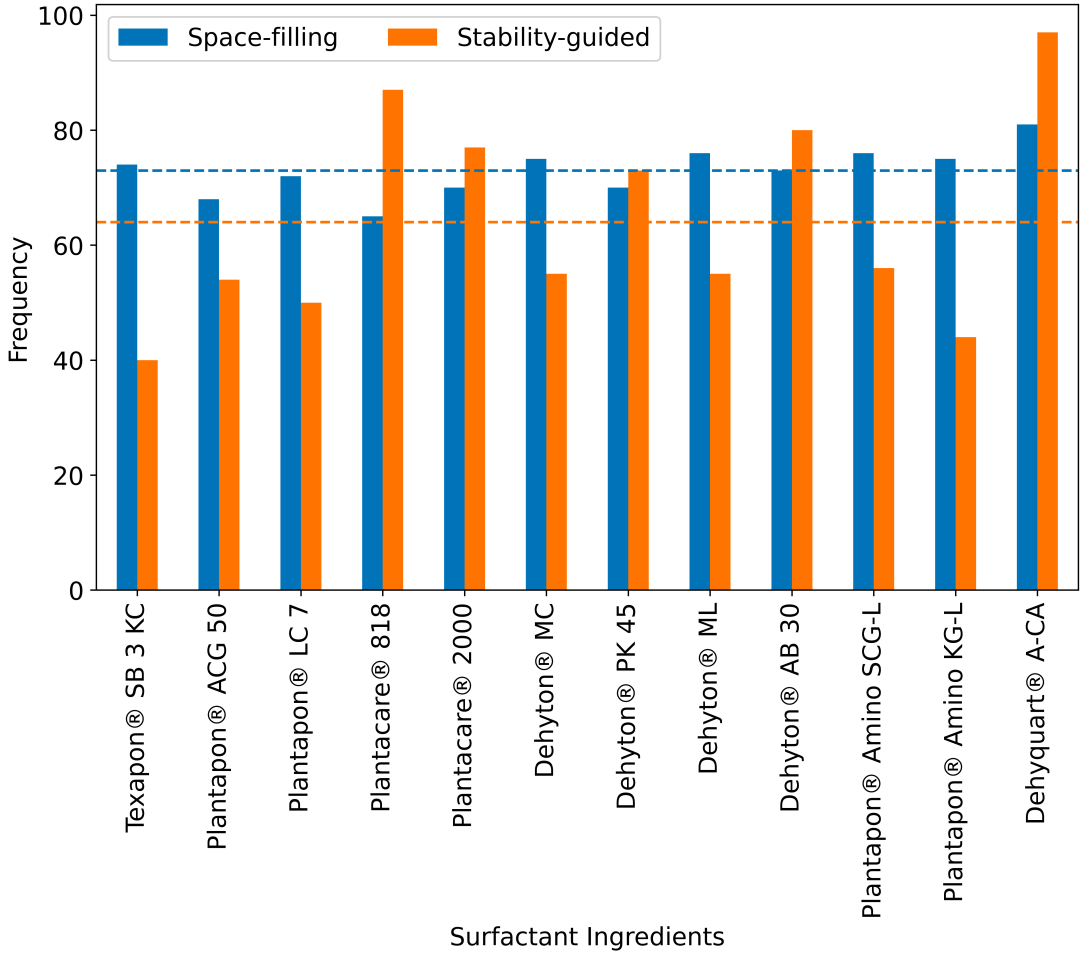

Figure 4. Spread of surfactants across the formulations dataset, sub-divided by the purely space-filling designs and stability-guided designs for the difficult-to-formulate sub-systems.

Looking at the selection of surfactants, polymer, and thickener covers the qualitative variables in our design problem; in Figure 5 we show the coverage of our quantitative design variables – ingredient concentrations (w/w%). As stated in Step (2) of Scheme 1, we aimed for the surfactants to have a distribution between 8 – 13 w/w%, conditioning polymers 1 – 3 w/w%, and thickeners 1 – 5 w/w%. We observe a good distribution of concentrations across all the ingredients, where the median and interquartile ranges (IQR) are given by the dashed lines on

the violin plots in Figure 5. Therefore, our full formulations design space has been represented in the generated dataset. We only note that for some ingredients, namely the very viscous ones, we exceeded the suggested concentration bounds for experimental reasons, however, this is acceptable and still informative towards developing property prediction models.

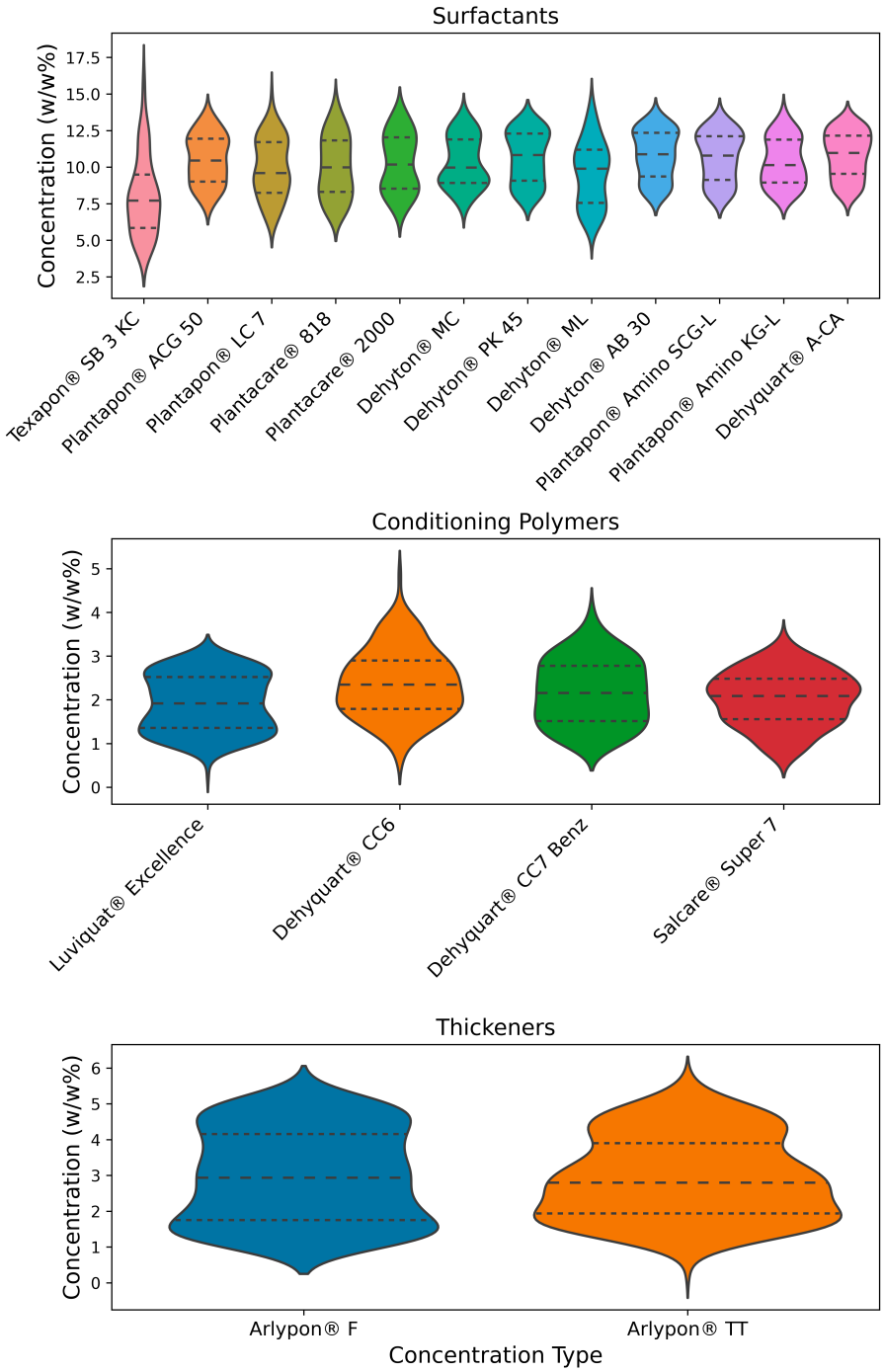

Figure 5. Distribution of formulation ingredient concentrations across the surfactants, conditioning polymers, and thickeners, which had target ranges of 8 – 13, 1 – 3, and 1 – 5 w/w%, respectively. The median concentration and interquartile ranges are shown on the plots. Where the target ranges have been exceeded this is due to viscous liquid handling challenges.

Phase Stability Classifiers and Chemical Interpretability

We now assess in Figure 6 the quality of the phase stability classifiers trained over the complete set of experimental data for the three difficult-to-formulate sub-systems. The receiver operating characteristic (ROC) curves in Figure 6 show the performance of the classification models at all different classification thresholds and the area under this curve (ROC AUC) provides an aggregate measure of the classifier's performance. Additionally, we have the class-weighted F_1 scores for the three classifiers. Both of these metrics go from 0 to 1 and whilst what constitutes a good score may be field or subject-dependent, typically, anything above 0.8 is good and above 0.9 is an excellent classifier. Given our relatively limited experimental budget and the high-dimensionality of this formulations case-study, we have developed highly predictive stability classifiers.

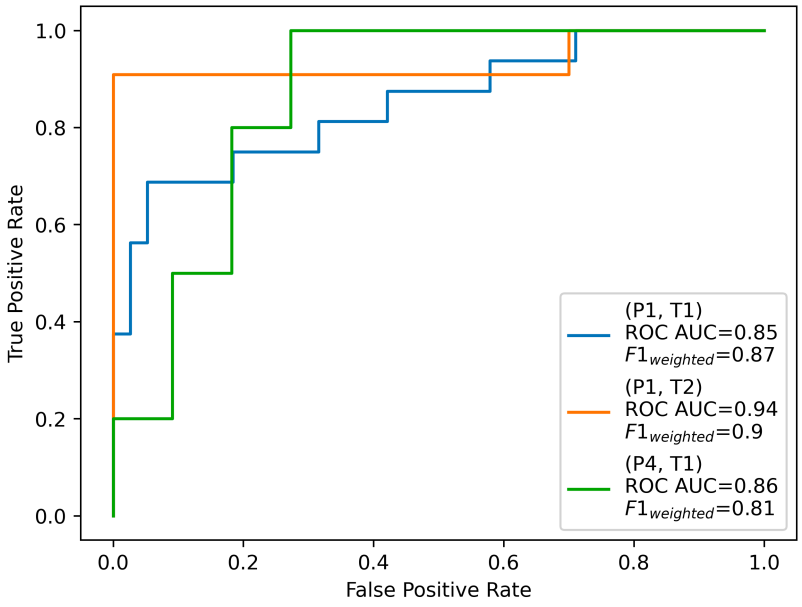

Figure 6. Receiver operating characteristic (ROC) curves for the best phase stability classifiers for each of the three difficult-to-formulate sub-systems.

Since we have trained strong phase stability classifiers and used a chemically interpretable representation for the surfactants, we can now draw reliable scientific insights from the results presented in Figure 7, showing feature importances and explanations for the (P_1, T_1) sub-system. These results for the (P_1, T_2) and (P_4, T_1) sub-systems are presented in Figure S3. The results in Figure S2 show that across all three sub-systems the best performing stability classifier was a random forest. This has the beneficial property that we can directly compute feature importances for tree-based models (Breiman, 2001), as shown in Figure 7a (and Figure

S3). These results are computed based on the decrease in model performance if a particular feature is removed. Another popular method in the field of ML interpretability is the use of SHAP (SHapley Additive exPlanations) (Lundberg and Lee, 2017). These values show how each feature affects the final prediction (Lundberg et al., 2020). SHAP is based on the magnitude of feature attributions. Feature importances and SHAP values are different measures, but it is interesting to note that the order of features presented in both Figures 7a and 7b are very similar. In both cases, the concentration of thickener is the most important factor governing phase stability, with Figure 7b showing less thickener is better for preparing stable formulations. These results are directly interpretable for the chemist or formulator, as we have attributed the stability (or instability) to surfactant functional groups, or polymer and thickener concentrations. Furthermore, as shown for an illustrated sample in Figure S4, SHAP can also provide feature attributions on a sample-by-sample basis for a deeper investigation of a formulation's properties. These *a posteriori* analyses can aid in developing novel formulated products. Further discussion linking the chemistry of the functional groups to the phase stability results is out of the scope of this work and will be treated in a future discussion.

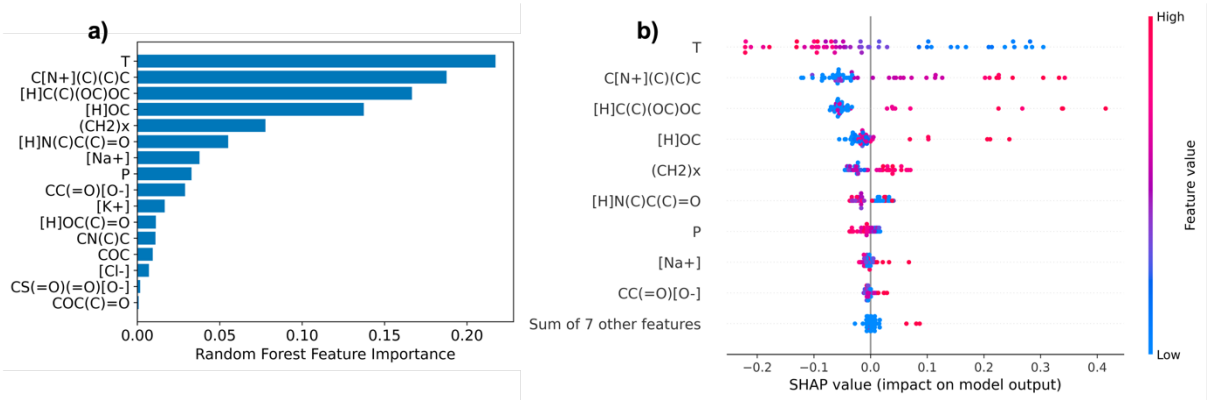

Figure 7. Chemical interpretability of the phase stability classifier for sub-system (P_1, T_1) via (a) random forest feature importances; and (b) SHAP feature explanations.

Conclusions

We developed a weighted-space filling design for liquid formulations built on restricting MaxProQQ designs to stable predicted regions of difficult-to-formulate sub-systems, where a sub-system looked at a fixed polymer and thickener combination. We produced a chemically interpretable featurisation by considering the functional groups present in our surfactant ingredients. Future work would be to extend the molecular representation also to the macromolecular ingredients. We successfully trained highly predictive phase stability classifiers for three difficult-to-formulate sub-systems. We used these classifiers to guide our

experiments to regions of stability. For the other five sub-systems, we used purely space-filling designs to better cover the entire design space. However, we discuss through the use of a tuneable phase stability cut-off in our ML-guided DoE method how we balanced exploration of our design space with exploitation towards feasible regions for the difficult-to-formulate sub-systems too. Therefore, the overall spread and coverage of our designs show satisfactory properties – a relatively uniform spread of ingredients and coverage of the entire concentration ranges we were interested in. The resulting experimental dataset from this work, therefore, represents the full design space from the set of available ingredients and suggested concentrations from our industry partner, which can now be used to develop property prediction models to accelerate formulation design.

Code Availability

The R script developed for this work, as well as Jupyter notebook to train the phase stability classifiers, is available at https://github.com/sustainable-processes/formulations_ML-DoE.

Acknowledgements

The project was co-funded by UKRI Program Grant Chembots: Digital-Chemical-Robotics to Convert Code to Molecules and Complex Systems (EP/S019472). AC is grateful to BASF for co-funding his PhD studentship. The project was co-funded by National Research Foundation (NRF), Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) program as a part of the Cambridge Centre for Advanced Research and Education in Singapore Ltd (CARES). Experiments for this study were performed at IMRE, A*STAR in the group of Prof. Kedar Hippalgaonkar; experimental work is described in detail elsewhere.

References

Abooali, D., Soleimani, R., 2023. Structure-based modeling of critical micelle concentration (CMC) of anionic surfactants in brine using intelligent methods. Sci Rep 13, 13361. https://doi.org/10.1038/s41598-023-40466-1

Ba, S., Myers, W.R., Brenneman, W.A., 2015. Optimal Sliced Latin Hypercube Designs. Technometrics 57, 479–487. https://doi.org/10.1080/00401706.2014.957867

Bagajewicz, M., Hill, S., Robben, A., Lopez, H., Sanders, M., Sposato, E., Baade, C., Manora, S., Hey Coradin, J., 2011. Product design in price-competitive markets: A case study of a skin moisturizing lotion. AIChE Journal 57, 160–177. https://doi.org/10.1002/aic.12242

Bernardo, F.P., Saraiva, P.M., 2005. Integrated process and product design optimization: a cosmetic emulsion application, in: Computer Aided Chemical Engineering. Elsevier, pp. 1507–1512. https://doi.org/10.1016/S1570-7946(05)80093-8

Bin Sulaiman, R., Schetinin, V., Sant, P., 2022. Review of Machine Learning Approach on Credit Card Fraud Detection. Hum-Cent Intell Syst 2, 55–68. https://doi.org/10.1007/s44230-022-00004-0

Bowman, V.E., Woods, D.C., 2013. Weighted space-filling designs. Journal of Simulation 7, 249–263. https://doi.org/10.1057/jos.2013.8

Bray, S.A., Lucas, X., Kumar, A., Grüning, B.A., 2020. The ChemicalToolbox: reproducible, user-friendly cheminformatics analysis on the Galaxy platform. J Cheminform 12, 40. https://doi.org/10.1186/s13321-020-00442-7

Breiman, L., 2001. Random Forests. Machine Learning 45, 5–32. https://doi.org/10.1023/A:1010933404324

Cao, L., Russo, D., Felton, K., Salley, D., Sharma, A., Keenan, G., Mauer, W., Gao, H., Cronin, L., Lapkin, A.A., 2021a. Optimization of Formulations Using Robotic Experiments Driven by Machine Learning DoE. Cell Reports Physical Science 2, 100295.

Cao, L., Russo, D., Lapkin, A.A., 2021b. Automated robotic platforms in design and development of formulations. AIChE J 67, e17248.

Cao, L., Russo, D., Matthews, E., Lapkin, A., Woods, D., 2023. Computer-aided design of formulated products: A bridge design of experiments for ingredient selection. Computers & Chemical Engineering 169, 108083. https://doi.org/10.1016/j.compchemeng.2022.108083

Chitre, A., Cheng, J., Ahamed, S., Querimit, R.C.M., Zhu, B., Wang, K., Wang, L., Hippalgaonkar, K., Lapkin, A.A., 2023. pHbot: Self-Driven Robot for pH Adjustment of Viscous Formulations via Physics-informed-ML. Chemistry Methods e202300043. https://doi.org/10.1002/cmtd.202300043

Chitre, A., Querimit, R.C.M., Rihm, S.D., Karan, D., Zhu, B., Wang, K., Wang, L., Hippalgaonkar, K., Lapkin, A.A., 2024. Accelerating Formulation Design via Machine Learning: Generating a High-throughput Shampoo Formulations Dataset. Under review

Consonni, V., Todeschini, R., 2010. Molecular Descriptors, in: Puzyn, T., Leszczynski, J., Cronin, M.T. (Eds.), Recent Advances in QSAR Studies, Challenges and Advances in Computational Chemistry and Physics. Springer Netherlands, Dordrecht, pp. 29–102. https://doi.org/10.1007/978-1-4020-9783-6_3

Conte, E., Gani, R., Ng, K.M., 2011. Design of formulated products: A systematic methodology. AIChE J. 57, 2431–2449. https://doi.org/10.1002/aic.12458

Ertl, P., 2017. An algorithm to identify functional groups in organic molecules. J Cheminform 9, 1–7. https://doi.org/10.1186/s13321-017-0225-z

Gani, R., 2004. Chemical product design: challenges and opportunities. Computers & Chemical Engineering 28, 2441–2457. https://doi.org/10.1016/j.compchemeng.2004.08.010

Gani, R., Ng, K.M., 2015. Product design – Molecules, devices, functional products, and formulated products. Computers & Chemical Engineering 81, 70–79. https://doi.org/10.1016/j.compchemeng.2015.04.013

Ghiringhelli, L.M., Vybiral, J., Levchenko, S.V., Draxl, C., Scheffler, M., 2015. Big Data of Materials Science: Critical Role of the Descriptor. Phys. Rev. Lett. 114, 105503. https://doi.org/10.1103/PhysRevLett.114.105503

Jessop, P.G., Ahmadpour, F., Buczynski, M.A., Burns, T.J., Green II, N.B., Korwin, R., Long, D., Massad, S.K., Manley, J.B., Omidbakhsh, N., Pearl, R., Pereira, S., Predale, R.A., Sliva, P.G., VanderBilt, H., Weller, S., Wolf, M.H., 2015. Opportunities for greener

alternatives in chemical formulations. Green Chem. 17, 2664–2678. https://doi.org/10.1039/C4GC02261K
Johnson, M.E., Moore, L.M., Ylvisaker, D., 1990. Minimax and maximin distance designs. Journal of Statistical Planning and Inference 26, 131–148. https://doi.org/10.1016/0378-3758(90)90122-B
Joseph, V.R., Gul, E., Ba, S., 2020. Designing computer experiments with multiple types of factors: The MaxPro approach. Journal of Quality Technology 52, 343–354. https://doi.org/10.1080/00224065.2019.1611351
Joseph, V.R., Gul, E., Ba, S., 2015. Maximum projection designs for computer experiments. Biometrika 102, 371–380. https://doi.org/10.1093/biomet/asv002
Kelly, C.L., 2023. Addressing the sustainability challenges for polymers in liquid formulations. Chem. Sci. 1–6. https://doi.org/10.1039/D3SC90086J
Kim, C., Chandrasekaran, A., Huan, T.D., Das, D., Ramprasad, R., 2018. Polymer Genome: A Data-Powered Polymer Informatics Platform for Property Predictions. J. Phys. Chem. C 122, 17575–17585. https://doi.org/10.1021/acs.jpcc.8b02913
Kronberg, B., Holmberg, K., Lindman, B., 2014. Surface Chemistry of Surfactants and Polymers. John Wiley & Sons, Ltd, Chichester, UK. https://doi.org/10.1002/9781118695968
Kuenneth, C., Ramprasad, R., 2023. polyBERT: a chemical language model to enable fully machine-driven ultrafast polymer informatics. Nat Commun 14, 4099. https://doi.org/10.1038/s41467-023-39868-6
Lin, T.-S., Coley, C.W., Mochigase, H., Beech, H.K., Wang, W., Wang, Z., Woods, E., Craig, S.L., Johnson, J.A., Kalow, J.A., Jensen, K.F., Olsen, B.D., 2019. BigSMILES: A Structurally-Based Line Notation for Describing Macromolecules. ACS Cent. Sci. 5, 1523–1531. https://doi.org/10.1021/acscentsci.9b00476
Lundberg, S.M., Erion, G., Chen, H., DeGrave, A., Prutkin, J.M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., Lee, S.-I., 2020. From local explanations to global understanding with explainable AI for trees. Nat Mach Intell 2, 56–67. https://doi.org/10.1038/s42256-019-0138-9
Lundberg, S.M., Lee, S.-I., 2017. A Unified Approach to Interpreting Model Predictions. Presented at the 31st Conference on Neural Information Processing Systems, Long Beach CA, USA, pp. 1–10.
Martín, M., Martínez, A., 2013. A methodology for simultaneous process and product design in the formulated consumer products industry: The case study of the detergent business. Chemical Engineering Research and Design 91, 795–809. https://doi.org/10.1016/j.cherd.2012.08.012
McKay, M.D., Beckman, R.J., Conover, W.J., 1979. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. Technometrics 21, 239–245.
Moriwaki, H., Tian, Y.-S., Kawashita, N., Takagi, T., 2018. Mordred: a molecular descriptor calculator. J Cheminform 10, 1–14.
Morris, M.D., Mitchell, T.J., 1993. Exploratory designs for computational experiments. Journal of Statistical Planning and Inference 43, 381–402.
Narayanan, H., Dingfelder, F., Condado Morales, I., Patel, B., Heding, K.E., Bjelke, J.R., Egebjerg, T., Butté, A., Sokolov, M., Lorenzen, N., Arosio, P., 2021. Design of Biopharmaceutical Formulations Accelerated by Machine Learning. Mol. Pharmaceutics 18, 3843–3853. https://doi.org/10.1021/acs.molpharmaceut.1c00469
O'Boyle, N.M., Banck, M., James, C., Vandermeersch, T., Hutchnison, G., 2011. Open Babel: An open chemical toolbox. Journal of Cheminformatics 3, 14.

Öztürk, H., Ozkirimli, E., Özgür, A., 2016. A comparative study of SMILES-based compound similarity functions for drug-target interaction prediction. BMC Bioinformatics 17, 128. https://doi.org/10.1186/s12859-016-0977-x

Qian, P.Z.G., 2012. Sliced Latin Hypercube Designs. Journal of the American Statistical Association 107, 393–399. https://doi.org/10.1080/01621459.2011.644132

Qin, S., Jin, T., Van Lehn, R.C., Zavala, V.M., 2021. Predicting Critical Micelle Concentrations for Surfactants Using Graph Convolutional Neural Networks. J. Phys. Chem. B 125, 10610–10620. https://doi.org/10.1021/acs.jpcb.1c05264

Schwartz, J., Awale, M., Reymond, J.-L., 2013. SMIfp (SMILES fingerprint) Chemical Space for Virtual Screening and Visualization of Large Databases of Organic Molecules. J. Chem. Inf. Model. 53, 1979–1989. https://doi.org/10.1021/ci400206h

Seddon, D., Müller, E.A., Cabral, J.T., 2022. Machine learning hybrid approach for the prediction of surface tension profiles of hydrocarbon surfactants in aqueous solution. Journal of Colloid and Interface Science 625, 328–339. https://doi.org/10.1016/j.jcis.2022.06.034

Taifouris, M., Martín, M., Martínez, A., Esquejo, N., 2020. Challenges in the design of formulated products: multiscale process and product design. Current Opinion in Chemical Engineering 27, 1–9. https://doi.org/10.1016/j.coche.2019.10.001

Tran, T.C., Dang, T.K., 2021. Machine Learning for Prediction of Imbalanced Data: Credit Fraud Detection, in: 2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM). Presented at the 2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM), IEEE, Seoul, Korea (South), pp. 1–7. https://doi.org/10.1109/IMCOM51814.2021.9377352

Vidal, D., Thormann, M., Pons, M., 2005. LINGO, an Efficient Holographic Text Based Method To Calculate Biophysical Properties and Intermolecular Similarities. J. Chem. Inf. Model. 45, 386–393. https://doi.org/10.1021/ci0496797

Wigh, D.S., Goodman, J.M., Lapkin, A.A., 2022. A review of molecular representation in the age of machine learning. WIREs Comput Mol Sci 12, 1–19.

Yang, K., Swanson, K., Jin, W., Coley, C., Eiden, P., Gao, H., Guzman-Perez, A., Hopper, T., Kelley, B., Mathea, M., Palmer, A., Settels, V., Jaakkola, T., Jensen, K., Barzilay, R., 2019. Analyzing Learned Molecular Representations for Property Prediction. J. Chem. Inf. Model. 59, 3370–3388. https://doi.org/10.1021/acs.jcim.9b00237

Yap, C.W., 2011. PaDEL-descriptor: An open source software to calculate molecular descriptors and fingerprints. J. Comput. Chem. 32, 1466–1474. https://doi.org/10.1002/jcc.21707

Supplementary Information for Machine Learning-Guided Space-filling Designs for High Throughput Liquid Formulations Development

Training a Phase Stability Classifier

Our formulations dataset, as featurised by the method shown in Figure 2, represents a structured, tabular dataset. We have a two-class classification problem – predict whether a formulation is stable (1) or not (0). We firstly trained and tested three different ML models: logistic regression (logreg), Naïve Bayes (NB), and a decision tree (DT) using default scikit-learn hyperparameters to develop a baseline performance for our phase stability classifier. We see from Figure S2 that we are able to improve on this baseline, as expected, when we tune more complicated models. We assessed the performance of the classifier via two metrics: i) area under the receiver operating characteristic curve (ROC AUC); and ii) a class-weighted F_1 score. The results of these baseline classifiers is shown in Figure S1.

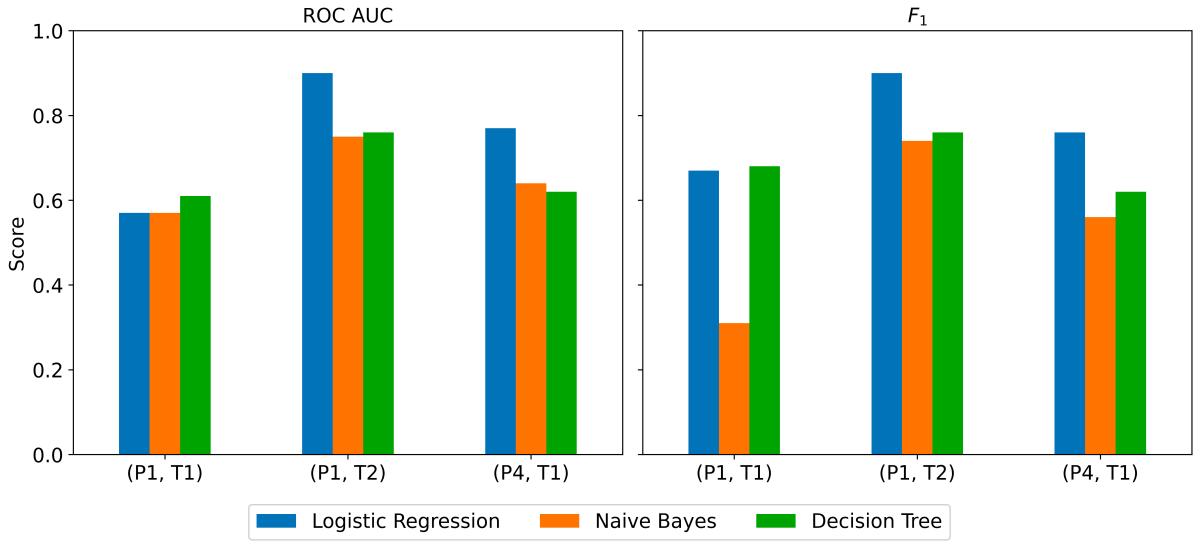

Figure S1. Comparison of three different ML models to establish a baseline performance for training a phase stability classifier over the different formulations sub-systems.

Given the relatively small dataset size, we concluded that deep learning models like neural networks were inappropriate for this classification task. Instead, we focused our efforts on three particular ML models: random forests (RF), support vector machines (SVM/SVC), and XGBoost (XGB). RF and XGB in particular are ensemble methods which typically work very

well on this type of data – structured and tabular. We focused on extensive hyperparameter tuning for these models using k-fold cross-validation (with $k = 6$) and three different methods: GridSearchCV, RandomSearchCV and BayesOpt (BO), as shown by the results in Figure S2. For each of the ML models, we tuned the following hyperparameters:

- RF: `n_estimators`, `min_samples_split`, `min_samples_leaf`, `max_features`, `class_weight`
- SVC: `C` (regularisation parameter), `kernel`, `degree` (if polynomial kernel), `class_weight`
- XGB: `learning_rate`, `n_estimators`, `max_depth`, `min_child_weight`, `gamma`, `colsample_bytree`

Further details can be found in the Jupyter Notebook on the linked GitHub repository (https://github.com/sustainable-processes/formulations_ML-DoE) for training the phase stability classifier. We note, for the SVC we only used the grid search method, as we could exhaustively enumerate all our hyperparameter combinations.

Presented in Figure S2 are the ROC AUC and F_1 scores as evaluated on the test-set for RF, SVC, and XGB models trained with different hyperparameter search strategies. We used a 75:25 train:test split. This data is presented for the phase stability classifier trained on the full experimental data for each sub-system. Here, RFs are the best-performing model across each sub-system. However, Tables S1-3 show that this is not always the case. An SVC or XGB may also be the best performing model at different iterations of the DoE. Figure S2 shows the ML performance differs in an unpredictable way based on the hyperparameter search strategy. We cannot pick a single best method and therefore, we exhaustively try all the tools at our disposal to ultimately develop the most predictive models, whose ROC curves are shown in Figure 6.

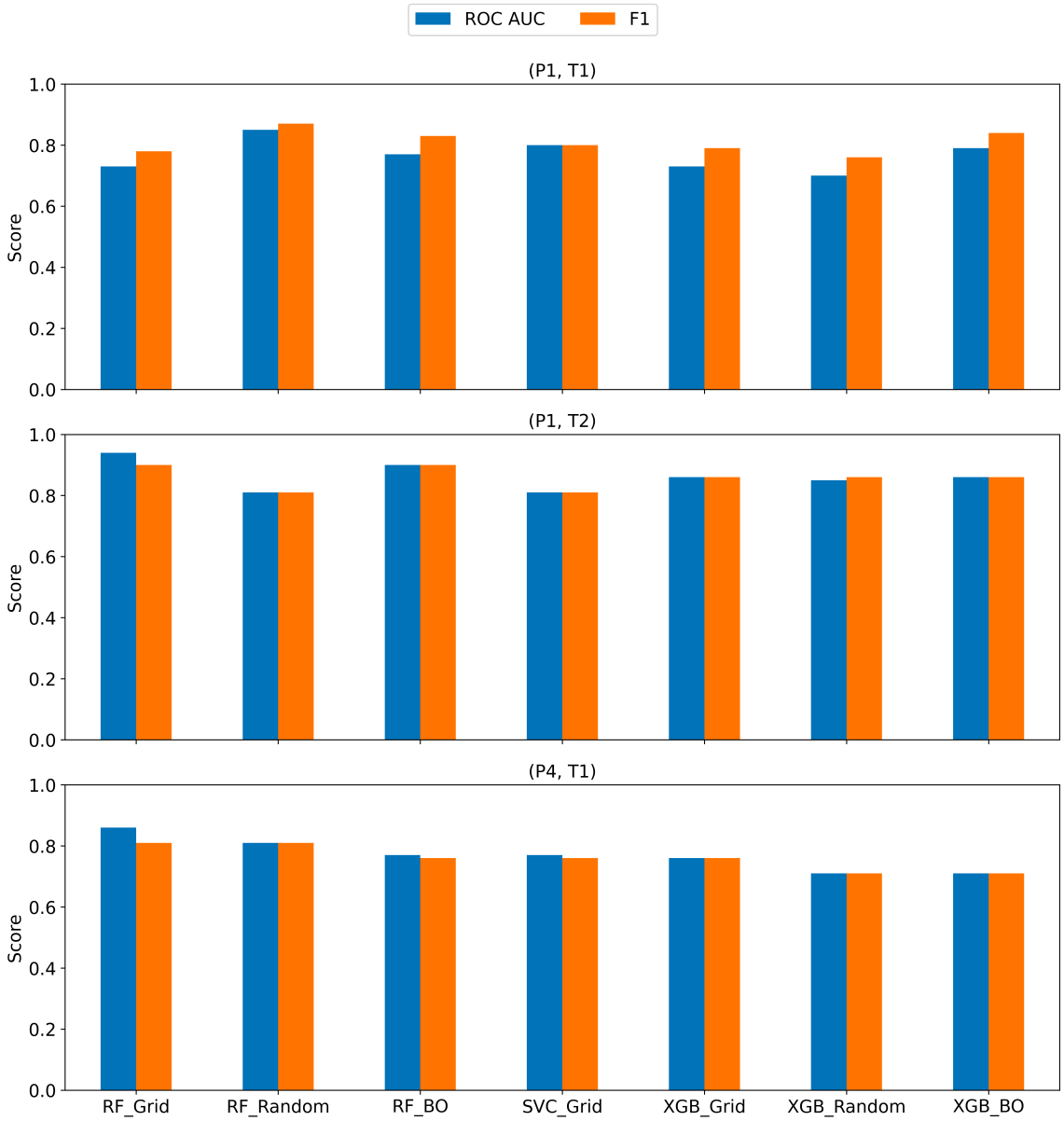

Figure S2. Results from hyperparameter tuning across three different ML models (Random Forest, Support Vector Classifier, XGBoost) using k-fold cross validation ($k = 6$) and three different hyperparameter search methods (`GridSearchCV, RandomSearchCV, BayesOpt`).

Phase Stability-guided Design of Experiments

Complementary to Figure 3 in the main text, Tables S1 – S3 provide a tabular summary of the phase stability-guided DoE results, additionally including the best performing ML model at each iteration of the DoE, along with its evaluation metrics (ROC AUC, F_1).

Table S1. Phase-stability guided DoE results for (P_1, T_1)

Round	ML Model & Performance (ROC, F_1 Scores)	Phase stability cut-off (top % stable or lowest probability of stability)	# Stable in Round	Overall % Stable
0	N/A	N/A	17/96	17.7
1	Random Forest (0.62, 0.73)	Top 20% (0.29)	7/24	20.0
2	Random Forest (0.83, 0.78)	Top 10% (0.38)	3/24	18.8
3	Random Forest (0.83, 0.87)	Top 5% (0.57)	19/36	25.0
4	XGBoost (0.80, 0.82)	(Top 3.7%) 0.80	19/36	30.1

Table S2. Phase-stability guided DoE results for (P_1, T_2)

Round	ML Model & Performance (ROC, F_1 Scores)	Phase stability cut-off (top % stable or lowest probability of stability)	# Stable in Round	Overall % Stable
0	N/A	N/A	8/36	22.2
1	SVC (1.00, 1.00)	(Top 8.4%) 0.60	11/24	31.7
2	SVC (0.96, 0.93)	(Top 0.13%) 0.75	24/24	51.1

Table S3. Phase-stability guided DoE results for (P_4, T_1)

Round	ML Model & Performance (ROC, F_1 Scores)	Phase stability cut-off (top % stable or lowest probability of stability)	# Stable in Round	Overall % Stable
0	N/A	N/A	13/48	27.1
1	Random Forest (0.85, 0.91)	(Top 14.8%) 0.60	27/36	47.6

Chemical Interpretability

Figure 7 in the main text discusses the random forest feature importances and SHAP feature explanations for the (P_1, T_1) phase stability classifier. Here a similar analysis is presented for the other two difficult-to-formulate sub-systems in Figure S3.

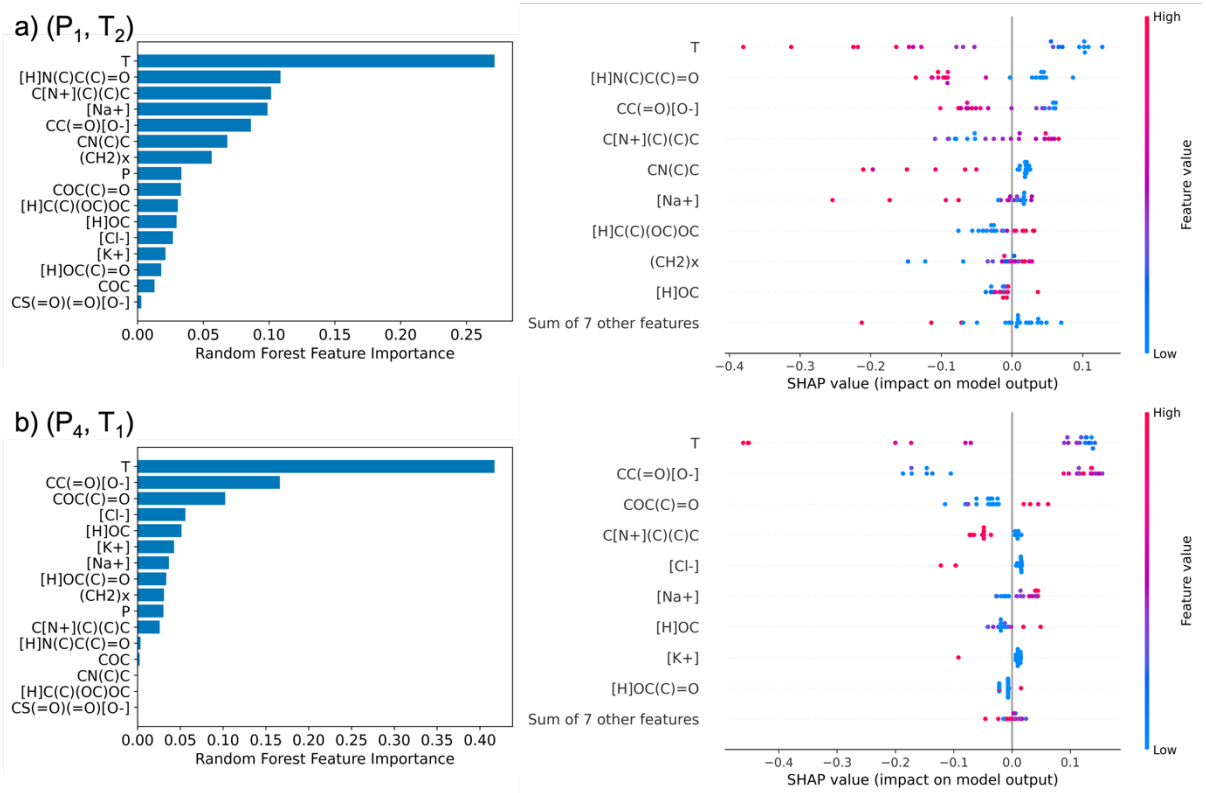

Figure S3. Feature importances and SHAP feature explanations for the phase stability classifiers for formulation sub-systems: (a) (P_1, T_2) and (b) (P_4, T_1).

Additionally, SHAP can provide feature explanations for individual formulations, as shown for a particular sample from sub-system (P_1, T_1) in Figure S3. Here the base value of 0.30 indicates the probability of any random sample within that sub-system being stable. This agrees with the result presented in Table S1. For the particular sample presented, it has a 0.19 probability of stability, and we can observe which individual features contribute to this prediction.

Figure S4. Explained phase stability prediction for an example formulation.