The Catechol Benchmark: Time-series Solvent Selection Data for Few-shot Machine Learning

Toby Boyne[1]*, Juan S. Campos[1], Becky D. Langdon[1], Jixiang Qing[1], Yilin Xie[1]
Shiqiang Zhang[1], Calvin Tsay[1], Ruth Misener[1], Daniel W. Davies[2], Kim E. Jelfs[2]
Sarah Boyall[3], Thomas M. Dixon[3], Linden Schrecker[3], Jose Pablo Folch[3]†
Department of Computing, Imperial College London, London, UK[1]
Department of Chemistry, Imperial College London, London, UK[2]
SOLVE Chemistry, London, UK[3]

Abstract

Machine learning has promised to change the landscape of laboratory chemistry, with impressive results in molecular property prediction and reaction retrosynthesis. However, chemical datasets are often inaccessible to the machine learning community as they tend to require cleaning, thorough understanding of the chemistry, or are simply not available. In this paper, we introduce a novel dataset for yield prediction, providing the first-ever transient flow dataset for machine learning benchmarking, covering over 1200 process conditions. While previous datasets focus on discrete parameters, our experimental set-up allow us to sample a large number of continuous process conditions, generating new challenges for machine learning models. We focus on solvent selection, a task that is particularly difficult to model theoretically and therefore ripe for machine learning applications. We showcase benchmarking for regression algorithms, transfer-learning approaches, feature engineering, and active learning, with important applications towards solvent replacement and sustainable manufacturing.

1 Introduction

Machine learning (ML) and artificial intelligence (AI) have showcased enormous potential in empowering the world of the natural sciences: from famous examples such as AlphaFold for protein predictions [1], to fusion reactor control [2], disease detection [3], battery design [4], and material discovery [5], among many more. However, we seldom see the machine learning community benchmark new methods in physical science datasets, mostly due to the difficulty in cleaning real-world data, the need for interdisciplinary understanding to correctly benchmark, and most importantly, how expensive the data can be to produce, resulting in many datasets being locked behind closed doors by large companies.

AIchemy (`https://aichemy.ac.uk`) is an interdisciplinary UK hub with the mission of transforming the chemistry-AI interface via aiding the collaboration of chemists and AI researchers, as well as addressing gaps in data standards, curation, and availability for AI use. In partnership with SOLVE Chemistry (`https://www.solvechemistry.com`), we present a first important step into addressing the dataset gap with the introduction of a new and unique open dataset for benchmarking low-data machine learning algorithms for chemistry.

Solvent selection is one of the biggest challenges for chemical manufacturing, with solvents often being the main source of waste in the manufacturing process [6]. Increased regulation on solvents and a drive to making process manufacturing more sustainable led to an interest in the discovery of greener

*`t.boyne23@imperial.ac.uk` ; †`jose@solvechemistry.com`

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Track on Datasets and Benchmarks.

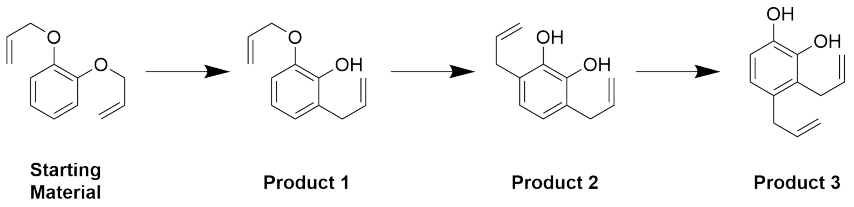


Figure 1: Data was gathered on the rearrangement of allyl substituted catechol. By subjecting the reaction mixture to high temperatures, we begin a cascade reaction forming multiple rearrangement products. We investigate the yield of the reaction for a range of different solvents. Product 1 was not observed and reacted immediately to form Product 2 and later 3.

solvents and for improved solvent replacement tools. However, most of the solvent replacement tools focus purely on learning unsupervised representations of solvents, with the hope that experimentalists can find solvents with similar properties to replace those with environmental concerns. A much stronger approach would consider the interaction of a variety of different solvents with a reaction of interest to directly predict reaction yields, in such a way that the best possible solvent can be selected according to a yield-sustainability trade-off.

Machine learning approaches have been shown to be a powerful tool for the prediction of chemical reaction conditions. Success has been reported in retro-synthesis [7, 8], condition recommendations [9], product predictions [10, 11], among others. While yield prediction has proven to be more difficult due to large inconsistencies in procedure and data reporting [12], we have still seen promising yield prediction results for smaller and more carefully curated datasets [13–16]. However, these datasets lack the continuous reaction conditions, such as temperature and residence time, that are required to scale-up processes to practical manufacturing conditions.

In this paper, we release the first machine-learning-ready transient flow dataset, a framework that allows for quick and efficient screening of continuous reaction conditions. We specifically provide yield data over the uni-molecular allyl substituted catechol reaction, shown in Figure 1, with dense measurements across the residence time, temperature, and solvent space. We answer the call for more flow chemistry reaction data [17], further showcase how this type of *kinetic data* poses new challenges to current machine learning methods for chemistry, and identify potential solutions.

1.1 Related works

Reaction datasets are common in chemistry research, but their suitability for machine learning benchmarking tends to be poor. This can be a result of improper formatting or documentation, incomplete information about reaction conditions or the experimental set-up, or the lack of machine readability, leading to limited usage by the ML community. However, some effort has been made to address this, with the biggest example being the creation of the Open Reaction Database (ORD) [18], a repository containing over 2M different reactions, many of which come from US patent data (USPTO) [19]. However, the dataset falls short in some aspects, in particular with respect to machine learning readiness and data inconsistencies across reactions.

ORDerly [12] allows for easy cleaning and preparation of ORD data, showing the promise of the dataset for forward and retro-synthetic prediction using transformers; however, it also shows that yield prediction cannot be done well due to data inconsistencies. Schwaller et al. [13] drew similar conclusions when using the USPTO dataset, stating that reaction conditions such as temperature, concentrations, and duration have a significant effect on yield. The assumption that every reaction in the dataset is optimized for reaction parameters proved too loose, resulting in inaccurate predictive models for yield, and highlighting the importance of creating datasets with full (including potentially sub-optimal) reaction conditions.

More relevant to our work, Perera et al. [20] introduced a dataset of 5760 Suzuki-Miyaura cross-coupling reactions, Ahneman et al. [21] introduced a dataset of 3956 Buchwald–Hartwig aminations, and Prieto Kullmer et al. [22] investigated screening additives for Ni-catalysed reactions, all for the purposes of yield prediction. The datasets have been used in the benchmarking of Gaussian processes and Bayesian neural networks [14], deep learning models [13], language-model-based embeddings

[16], data augmentation techniques [23], and Bayesian optimisation [15]. In each case, the datasets focus on discrete reaction variables, such as ligand, base, additives, or reactants at fixed temperatures and residence times. We are instead introducing a dataset rich in continuous reaction conditions (in our case temperature and residence time), as well as providing a pseudo-continuous representation of solvents themselves through the use of solvent mixtures.

Perhaps the closest example to our dataset is presented in Nguyen et al. [24], who used high-throughput experimentation to screen 12708 catalyst informatics on the oxidative coupling of methane. In this case, they do provide process conditions for temperature, reactant equivalents, and flow rates; however, they do so in a discretized manner, as opposed to our approach that produces denser continuous representations of variables. The dataset has been used in the context of benchmarking language models for yield prediction, where the variables are used to create prompts to generate LLM embeddings of reactions. Introduced by Ramos et al. [25], the LLM embeddings are used for Bayesian optimization in reaction space. Recently, Ranković and Schwaller [16] fine tune a subset of 1180 LLM embeddings to use as deep kernel GPs, achieving even more favorable performance.

More detailed and dense datasets including kinetic data usually have to be searched for in the research papers that originally published them, which are often accompanied by kinetic model fitting and benchmarking [26–28]. However, the datasets are seldom ML-ready, and tend to focus on variables which have predictable outcomes. In this work, we collect solvent data, which has a very large impact on the system's dynamics and is often very challenging to model theoretically, making it a particularly interesting instance for machine learning applications.

1.2 The dataset

The full dataset we collected for this project consists of 1227 data-points, with different reaction conditions, with the inputs being:

(1) A selection of two different solvents in which the reaction was carried out, and the corresponding ratio of the solvents in the mixture.

(2) The temperature at which the reaction was carried out.

(3) The residence time of the reaction, i.e., how long the reactants were subject to the reaction conditions applied.

The outputs consist of the yield of the starting material and the two observed products. We can further extract a dataset of single solvent data only. A visual summary of the data is provided in Figure 2.

We further expand the dataset by including previous measurements on the same reaction class collected on a similar molecule, which was reported when developing the solvent ramping technology [29]. The two datasets are detailed in Table 1, and can be downloaded from Kaggle [2]. We further include sustainability details for all the solvents screened in Table 2 according to the GSK guide [30].

Table 1: Summary of the datasets: solvent types, data sizes, output measurements, and presence of time-series data. SM = Starting Material.

Dataset	Subset	Data Points	Solvents	Output Yields	Time-Series
Allyl Substituted Catechol	Solvent Mixtures	1227	24	SM + 2 Products	✓
	Single Solvents	656	24	SM + 2 Products	✓
Allyl Phenyl Ether	Solvent Mixtures	283	11	SM + 1 Product	×

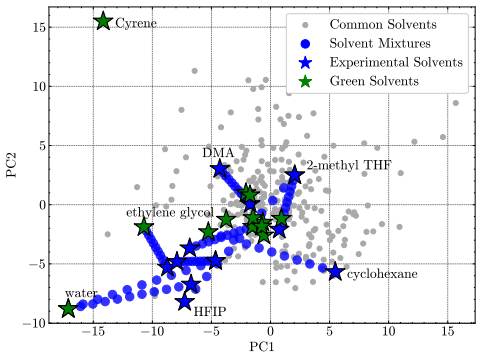
2 Dataset collection and techniques

This section provides general descriptions of the data collection techniques, transient flow, analytical analysis, deconvolution, and how active learning was used for ramp selection.

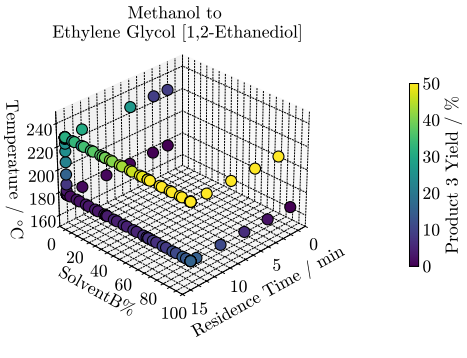
[2]Dataset: `https://www.kaggle.com/datasets/aichemy/catechol-benchmark/`.
Code: `https://github.com/jpfolch/catechol_solvent_selection`.

Table 2: List of screened solvents and their classification. For more details see Henderson et al. [30].

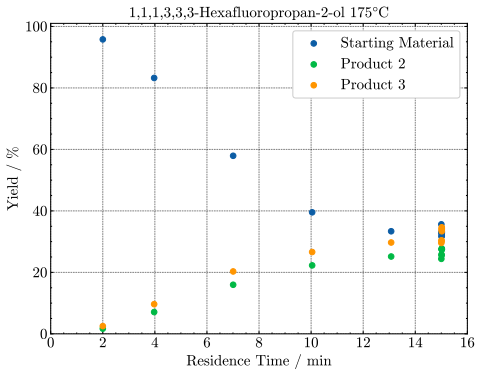
Classification	Solvents
Green	Ethylene Glycol; IPA; Water; Ethanol; Cyrene; Ethyl Acetate; DMC
Situation dependent	Methanol; 2-MeTHF; Cyclohexane; Acetonitrile; Acetic Acid; 2-Butanone; t-Butanol
Needs replacement	Diethyl Ether; DMA; THF; MTBE
Not on GSK guide	HFIP; 2,2,2-TFE; Decanol; Methyl Propionate; Ethyl ℓ-Lactate



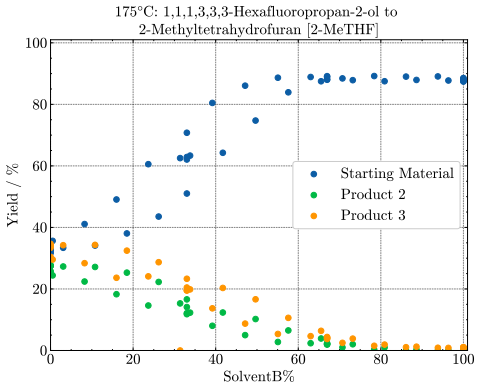
(a) ACS PCA representation of the space of solvents. We highlight the solvents we collected data for with green or blue stars, and show the mixture solvents as dots.



(b) Three-dimensional scatter plot showing an experimental run between two solvents. We see examples of residence time ramps, temperature ramps, and solvent ramps.



(c) Example of a residence time ramp under the HFIP solvent. We see how longer reaction time increases product yield.



(d) Example of a solvent ramp between HFIP and 2-MeTHF, exemplifying two of the challenges of the dataset: bias and heteroskedasticity.

Figure 2: Visual summary of the data set. (a) Showcases the solvent space covered. (b) A full 8h experimental run between two solvents. (c) A residence time ramp, showing the starting material and product yields. (d) A solvent ramp, showing the yields under solvent mixture conditions.

2.1 Transient flow and solvent ramping

Flow chemistry refers to a process in which the reaction is carried out in a continuous stream of reactant materials confined within tubing or narrow channels [31]. This technology offers an alternative to the traditional batch vessels typically used in chemical manufacturing and often presents benefits in areas such as safety, environmental impact, and scalability predictions [32]. Advantages include improved heat and mass transfer, more precise process control of the reaction conditions due to the smaller reaction volumes, and the ease of integration of online equipment and analytics [33].

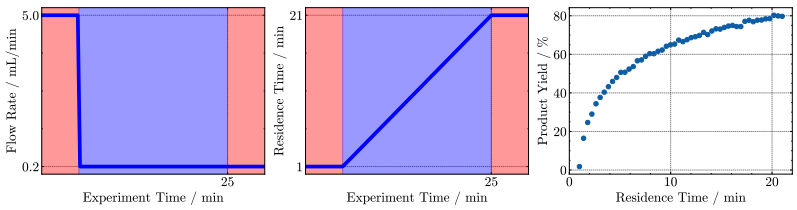


Figure 3: Example of a residence-time ramp in a transient flow reactor. (Left) We decrease the flow rate of the reactor to begin the experiments. (Middle) The residence time experienced by the flow at the point of measurement. (Right) Product yield mapped against residence time of measurements.

Transient flow chemistry is an emerging technology used for collecting large quantities of reaction data in a continuous system. The method differs from traditional steady-state flow chemistry techniques, as the reaction conditions are varied continuously during experimentation to screen a range of reaction conditions [34]. Due to the efficient mixing inherent to flow systems, when the reaction conditions are adjusted in a controlled manner, each individual part of the flow is subject to different reaction conditions (i.e., plug flow), resulting in the efficient collection of a series of data [35]. An example of a relationship that can be investigated using this technique is the effect of the reaction time (i.e., the residence time of the reaction) on the yield of the reaction. This can be done by changing the cumulative flow rate in the reactor. The flow rates of the system are initially set to correspond to a specific residence time, and a step change to lower flow rates is performed. This means that the plugs at the end of the reactor experiences a shorter residence time in the reactor, and each subsequent volume of flow, or 'plug,' will have a longer residence time, producing a continuous data series of increasing residence times [36]. A visualization of this process is given in Figure 3.

In a similar fashion, other variables can be investigated, such as temperature (varied by slowly ramping the reactor temperature) [37], and the equivalents of reagents in a reaction (by modifying the ratio of flow rates pumped from different reagent reservoirs) [38]. Reaction solvent is of particular interest in this dataset, following research interest in finding 'green solvents' as alternatives to traditional solvents [29, 39]. Solvent (mixtures) are treated as continuous variables, where the ratio of the two chosen solvents is varied using the ratio of the flow rates of the respective pumps to screen different solvent mixtures, and the changes in reaction yield are observed. Figure 2a shows all the solvent mixtures we were able to sample, as we effectively gather data between pure solvent pairs.

2.2 Solvent selection

In order to maximize the amount of information gathered from the data set, we used active learning. In particular, we trained a Gaussian process model on the Allyl Phenyl Ether dataset [29], which was the first published dataset that investigated solvent ramping using transient flow. We then selected a range of suitable available solvents [40] to create a set $\mathcal{S}$ and selected the solvents to query according to the entropy criterion:

$$s_{A,n+1}, s_{B,n+1} = \underset{s_A, s_B \in \mathcal{S}\times\mathcal{S}}{\arg\max}\, H(Y(s_A, s_B)|D_n), \tag{1}$$

where H is the GP's entropy, $D_n = \{X(s_{A,i}, s_{B,i})\}_{i=1}^{n}$ the set of solvent ramps gathered up to time n, and $Y(s_A, s_B)$ the vector of data points gathered during solvent ramping from s_A to s_B.

2.3 Data acquisition and preprocessing

Online analytical measurements were collected using high-performance liquid chromatography (HPLC) [41], sampled every two minutes. This allows quantitative yield measurements to be collected over the course of the reaction, which can then be related to the reaction conditions applied to each sample. These can be calculated for each variable since we know the reactor volume, the flow rates, the temperature, and the duration that each sample in the flow stream was subjected to particular reaction conditions. The residence time is calculated by considering the measurement time, t_m, the reactor volume, V, and the cumulative flow rate function, $F_c(t)$. We then solve the equation:

$$V = \int_{t_i}^{t_m} F_c(t)dt \tag{2}$$

to find the initial time the plug entered the reactor, t_i. From this we can then estimate all the reaction conditions, residence time, R_τ, solvent B percentage, $S_{B\%}$, and the average temperature, $\hat{T}$:

$$R_\tau = f_m - f_i; \quad S_{B\%} = \frac{F_B(t_i)}{F_c(t_i)}; \quad \hat{T} = \frac{1}{t_m - t_i} \int_{t_i}^{t_m} T_r(t) dt \tag{3}$$

where $F_B(t)$ is the flow rate of the solvent B pump at time t, and $T_r(t)$ is the temperature of the reactor at time t. We generally seek to make small, incremental changes in temperature and flow rates to obtain accurate measurements.

3 Machine Learning Benchmarks

In this section, we train a variety of machine learning models to investigate the performance of standard state-of-the-art models for this prediction task. In particular, we examine a large range of solvent featurization methods, and algorithms that have shown strong performance in the past.

3.1 Solvent featurization

Perhaps the most challenging, and most important, component of the benchmark problem we present is the solvent featurization process. This step asks how to represent each solvent (mixture), such that ML algorithms can extract suitable information for accurate predictions.

As our goal is to predict yield surfaces on unseen solvents, a featurization that allows for transfer of information between solvents is required. Diorazio et al. [40] introduced a dataset of 272 solvents, with over 100 features for each, and further provide a 5-dimensional PCA representation of the solvent space. A second representation uses measurable properties of solvents [42], allowing easy grouping of solvents by type, e.g., as esters, ethers, and alkanes.

Cheminformatic features of molecules [43], 'fragments', are created using count vectors indicating the number of times a specific functional group appears in the molecule (following group contribution theory). Rogers and Hahn [44] show that bit vectors indicating the presence of substructures in a molecule, coined 'molecular fingerprints', can be used for molecular property prediction. We test the concatenation of both vectors, known as 'fragprints' [45].

Finally, we investigate directly featurizing the reaction itself. This can be done, e.g., using the difference in sets containing molecular substructures in the starting materials and products [46, 47], known as differential reaction fingerprints (DRFP). Additionally, a reaction fingerprint can be learnt from larger open-source databases by using encoder-decoder neural networks [13], known as reaction fingerprints (RXNFP).

Featurizing solvent mixtures A further question of interest is that of how to represent mixtures of solvents. Given a pair of solvents and their respective featurizations S_A and S_B, we will initially take the naive approach of using a weighted mean: $S_{A\cup B}(b) = (1-b)S_A + bS_B$, where b is the proportion of solvent B in the mixture. However, this linear transition can be an oversimplification of the underlying chemistry [48], so we investigate learning a non-linear mapping in Section 3.3.

3.2 Regression

To evaluate the available machine learning tools for analyzing the dataset, we present a set of models for regression. We regress on the solvent mixture and single solvent datasets described in Table 1. We perform leave-one-out cross validation for all the models. Further details on the methods used and experimental details can be found in the appendix.

When creating the test set, we take the mean of all repeated observations to avoid over-penalizing models that predict these reaction conditions poorly. We also omit the reactions containing acetic acid due to unintended a side-reaction[3]; creating models that are robust to unexpected side reactions poses an interesting future challenge.

[3]The presence of acetic acid and high temperatures resulted in an unintended side-reaction of the expected product - possibly an esterification - as soon as it formed, showing very little yield of desired product but with high conversion. As such, we removed the affected results from our benchmark numbers.

Table 3: Regression performance on the full dataset. Mean squared error (MSE) and negative log predictive density (NLPD) are averaged across all leave-one-out data splits.

		Full data		Single solvent	
		MSE (↓)	NLPD (↓)	MSE (↓)	NLPD (↓)
Model	Featurization				
Baseline GP		***0.011***	-5.381	0.014	-5.044
GP	acs	0.016	-4.161	0.017	-4.053
	drfps	0.015	-4.937	0.017	-4.028
	fragprints	0.013	-5.010	0.017	-4.481
	spange	***0.011***	**-5.663**	***0.011***	**-5.793**
MLP	acs	0.014	-	***0.011***	-
	drfps	0.013	-	0.015	-
	fragprints	***0.011***	-	**0.010**	-
	spange	**0.010**	-	**0.010**	-
LLM	rxnfp	0.105	-	0.055	-
	chemberta	0.153	-	0.074	-
NODE	spange	-	-	0.055	-
EODE	spange	-	-	0.050	-3.339
LODE	spange	-	-	0.049	-3.235

Gaussian processes (GPs) are probabilistic, nonparametric models [49]. They are characterized by a covariance function, or *kernel*, that measures the similarity between a pair of inputs. These models provide uncertainty quantification, and perform well in low-data settings such as Bayesian optimization [50].

For our BaselineGP, we fit the data to the temperature and residence time inputs, ignoring the solvent, thus providing an improved proxy of one-hot encoding which has been shown to work well in low-data chemical regimes [51]. For the other GP methods, we transform the solvent into the featurized space, then use the Euclidean distance in that space as an input to an RBF kernel. In the appendix, we also evaluate a kernel over graphs, using pairwise shortest distances [52–54].

Neural networks are a now-ubiquitous tool for regression. We start with a simple multi-layer perceptron (MLP). We also fine-tune pretrained large language models (LLMs): ChemBERTa [55], a model pretrained on chemistry texts, and RXNFP [13], pretrained on chemical reaction classifications. This follows works that have shown LLM abilities to generalize across reactions using their string representations [56]. In the appendix we further show results for neural process architectures which mimic GPs through meta-learning approaches [57].

Latent ODE methods use neural networks to model the latent state and dynamics of an ordinary differential equation (ODE) [58, 59], directly representing the underlying kinetics of the reaction. We also include an explicit state variant (LODE) [60], and a neural ODE (NODE) with time-dependent dynamics.

The regression results in Table 3 show the strength of the Spange featurization, which uses parameters known to be important to solvent effects. Whilst MLPs have a slight edge over GPs in their mean square error performance, the latter is also able to provide uncertainty quantification. Using LLM embeddings leads to poor performance, as reported previously, however, the same work reports strong performance when the embeddings are fine-tuned [16] which is left to future work.

3.3 Gaussian process extensions

In Section 3.2, we report the performance of a selection of off-the-shelf machine learning algorithms for performing regression. However, many of these models fail to outperform the baseline model, which does not use any solvent information. We therefore propose some further GP models that can improve performance. Details of these models can be found in the appendix.

Kernel design must be performed carefully when creating GP models. A key issue with using the standard RBF kernel is that, for unseen solvents with featurizations very dissimilar to the solvents in

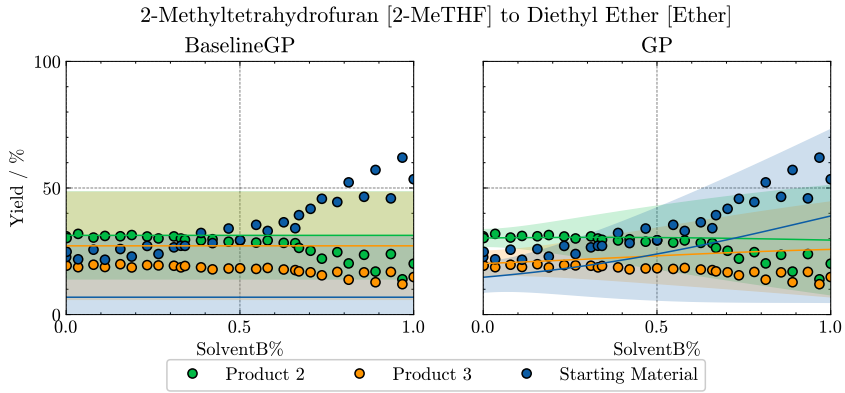


Figure 4: GP prediction on the yields of a solvent ramp, using Spange descriptors. We showcase a comparison between the baseline Gaussian process and a standard one. 2-MeTHF appears in another ramp, and so the model is confident about its predictions; as the proportion of Ether increases, so too does the model uncertainty.

Table 4: Regression performance of GP extensions on the catechol dataset.

		Full data		Single solvent	
		MSE (↓)	NLPD (↓)	MSE (↓)	NLPD (↓)
Model	Extension				
BaselineGP		0.011	-5.381	0.014	-5.044
GP		**0.011**	**-5.663**	0.011	-5.793
	Decomposed kernel	0.012	-5.455	**0.009**	**-6.091**
	Multitask GP	0.018	-2.885	0.011	-2.494
	Input warping	0.012	-4.781	0.011	-5.902

the train set, the GP will revert to the uninformative prior. We therefore propose decomposing the kernel into solvent and non-solvent components in an additive manner, similarly to Ru et al. [61].

Multitask kernels are able to learn correlations between outputs [62]. For example, the yields of the two products tend to be positively correlated with each other, and negatively correlated with the remaining starting material.

Nonstationary approaches allow modeling of search spaces that have changing lengthscale. For example, the rate of reaction is fastest in the first few minutes, and the solvent mixing may be nonlinear in the feature space as noted in Section 3.1. We therefore learn a warping of these two inputs, inspired by Snoek et al. [63] and Balandat et al. [64].

The results of these extensions are presented in Table 4. These show promising directions to improve regression on single solvents, but struggle to beat the simpler GP when introducing solvent ramps. We encourage the machine learning community to investigate further extensions, such as considering the heteroskedastic nature of the observation noise [65], or non-stationary kernels [66, 67].

3.4 Transfer learning

A key challenge in this dataset is the relatively low amount of data: where many modern ML approaches require large volumes of data, we only have 1227 observations of reaction conditions. To address this, we extend the training data by including results from the Ethyl dataset [29], which has a further 283 experiments. For this regression problem, we only predict the total product yield, since the Ethyl dataset only has one observed product.

We test the best performing regression models, the independent GPs and the MLP. For transfer of information across GPs, we use a multitask kernel corresponding to each reaction. For MLPs, we encode the reaction through a binary input. The baseline GP only uses the residence time and

Table 5: Regression performance with transfer learning from the Ethyl dataset.

		Catechol		Catechol + Ethyl	
Model	Featurization	MSE (↓)	NLPD (↓)	MSE (↓)	NLPD (↓)
BaselineGP		**0.023**	-1.331	0.023	-1.331
GP	spange	0.030	**-1.487**	**0.020**	**-1.506**
MLP	spange	0.027	-	0.031	-

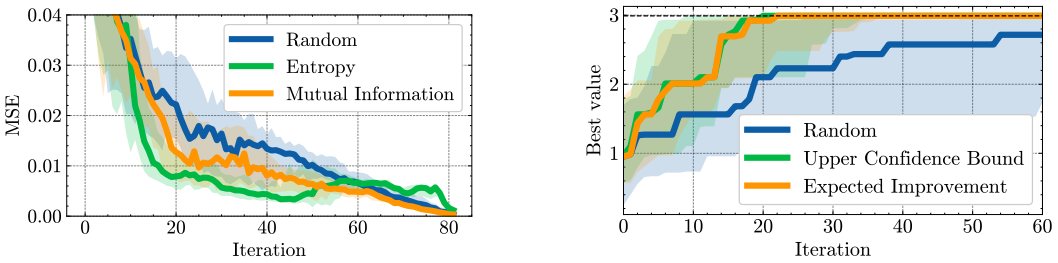
temperature, so cannot use the additional data. The results of this experiment are given in Table 5, demonstrating the utility of learning across multiple datasets.

3.5 Active learning and Bayesian optimization

One key application for machine learning in chemistry is to optimally design experimental conditions, with recent interest in transient flow applications [68–70]. In this section, we showcase how the dataset can be used to benchmark algorithms for design of experiments. For simplicity, we focus on the independent GP model with descriptors from Spange et al. [42]. We first explore *active learning* ideas in transient flow: we split the dataset into ramps and use the entropy and mutual information criteria [71] to select transient ramps sequentially, with the goal of maximizing information across the dataset, which we measure by MSE. Figure 5a shows that using the entropy criterion reduces MSE more initially, while using mutual information gives the best long-term performance.

We then benchmark on classical Bayesian optimization algorithms Expected Improvement [72] and Upper Confidence Bound [73]. We design an objective to maximize product yield and the selectivity of Product 2, while minimizing temperature and residence time, exemplifying conflicting objectives in the scale-up process. We allow the algorithms to query single points across the whole dataset, with the goal of identifying the optimal configuration in the fewest queries. Figure 5b shows the results. In this case, we observe that the algorithms are very quickly able to identify the optimum, usually after 20 iterations, outperforming a random search by a significant margin.

Finally we consider a multi-objective optimization benchmark, by considering a three dimensional objective function of trying to optimize yield, selectivity, and a green-score created from Table 2. We consider the solvent greenness by setting a value of 1.0 to every green solvent, -1.0 to every harmful solvent, and 0.0 to the remaining. For mixture solvent data-points we take a weighted average of their green scores. As benchmark metrics we consider three metrics of Pareto coverage: Euclidean generational distance (GD), inverted generational distance (IGD), and the maximum Pareto frontier error (MPFE) [74]. We present results in Table 6, where we benchmark Thompson Sampling with random scalarazations [75].



(a) Active learning on transient flow ramps.

(b) BO benchmarking across reaction space.

Figure 5: Results of benchmarking for active learning and BO. We initialize the GP with 5 random samples, and show results over 30 initializations. We report the median, 10th and 90th quantiles.

Table 6: Results of multi-objective optimization benchmarking over iterations. We compare Euclidean generational distance (GD), inverted generational distance (IGD), and the maximum Pareto frontier error (MPFE). Multi-objective optimization results in much stronger metrics that randomly sampling.

Iteration	GD (↓)		IGD(↓)		MPFE (↓)	
	Random	MOBO	Random	MOBO	Random	MOBO
1	0.199	**0.158**	0.397	**0.356**	0.453	**0.387**
25	0.107	**0.043**	0.177	**0.146**	0.421	**0.225**
50	0.081	**0.023**	0.126	**0.105**	0.388	**0.212**
75	0.069	**0.017**	0.106	**0.084**	0.385	**0.200**
100	0.054	**0.014**	0.092	**0.072**	0.285	**0.200**

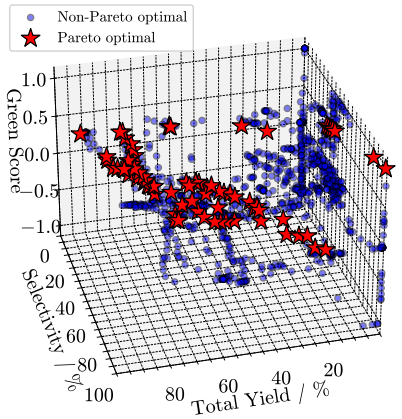


Figure 6: Visualization of empirical 3-dimensional Pareto front for the multi-objective Bayesian optimization benchmark.

4 Conclusions and future work

This paper introduces the first ML-ready transient flow reaction dataset, showcasing the dynamic nature of chemical reactions not fully considered in past datasets. We particularly focus on solvent selection and challenge of learning solvent effects. We benchmark a variety of regression algorithms and solvent featurizations, many of which have shown strong performance in chemistry applications before. We show that many algorithms struggle in our dynamic setting due to a variety of factors: non-stationarity, hetero-skedasticity, and most importantly the lack of a good solvent featurization method. However, we show current techniques can still lead to important improvements, and can be effective in active learning settings.

We call on the machine learning community to develop improved methods for chemical dynamical systems. Such methods need to be ready to be infused with prior chemical knowledge, either through priors or data-driven learning. However, the most important step we must first address is the lack of data - truly effective predictive models will require large understanding of chemistry that cannot be obtained from single datasets. For example, some solvents may enable side reactions even when present only at small concentrations; as we observed in this dataset for the case of acetic acid. The best possible representation over mixed solvents should therefore reflect even trace amounts of these solvents, and then consider not only yield predictions, but the probability of the reaction actually happening. We hope this work enables an important next step for many ML researchers, to develop even more intelligent chemistry models in the near future.

Limitations of this paper include the focus on only two reactions, and while we touch on a large amount of machine learning models, we only go in depth with Gaussian processes due to their suitability to the small data regime. Future work would include improvements and further research into deep learning models, more flexible Bayesian models such as Bayesian neural networks, and investigating further ways of encoding chemical information into models.

Acknowledgments

The authors thank the AI for Chemistry: AIchemy hub for funding (EPSRC grants EP/Y028759/1 and EP/Y028755/1). We would further like to thank the Engineering and Physical Sciences Research Council (grants EP/W003317/1 to RM&JCS, EP/X025292/1 to CT, RM, & JQ, StatML CDT EP/Y034813/1 and IConIC EP/X025292/1 to TB & BDL), a BASF/RAEng Research Chair in Data-Driven Optimisation to RM, a BASF/RAEng Senior Research Fellowship to CT, an Imperial College Department of Computing Scholarship to YX, and BASF SE, Ludwigshafen am Rhein funding to BDL, SZ, & TB. RM holds concurrent appointments as a Professor at Imperial and as an Amazon Scholar. This paper describes work performed at Imperial prior to joining Amazon and is not associated with Amazon.

References

[1] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. *nature*, 596(7873):583–589, 2021.

[2] Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de Las Casas, et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897): 414–419, 2022.

[3] Yukun Zhou, Mark A Chia, Siegfried K Wagner, Murat S Ayhan, Dominic J Williamson, Robbert R Struyven, Timing Liu, Moucheng Xu, Mateo G Lozano, Peter Woodward-Court, et al. A foundation model for generalizable disease detection from retinal images. *Nature*, 622 (7981):156–163, 2023.

[4] Jose Pablo Folch, Robert M Lee, Behrang Shafei, David Walz, Calvin Tsay, Mark van der Wilk, and Ruth Misener. Combining multi-fidelity modelling and asynchronous batch Bayesian optimization. *Computers & Chemical Engineering*, 172:108194, 2023.

[5] Paul Raccuglia, Katherine C Elbert, Philip DF Adler, Casey Falk, Malia B Wenny, Aurelio Mollo, Matthias Zeller, Sorelle A Friedler, Joshua Schrier, and Alexander J Norquist. Machine-learning-assisted materials discovery using failed experiments. *Nature*, 533(7601):73–76, 2016.

[6] David JC Constable, Conchita Jimenez-Gonzalez, and Richard K Henderson. Perspective on solvent use in the pharmaceutical industry. *Organic process research & development*, 11(1): 133–137, 2007.

[7] Pavel Karpov, Guillaume Godin, and Igor V Tetko. A transformer model for retrosynthesis. In *International conference on artificial neural networks*, pages 817–830. Springer, 2019.

[8] Igor V Tetko, Pavel Karpov, Ruud Van Deursen, and Guillaume Godin. State-of-the-art augmented NLP transformer models for direct and single-step retrosynthesis. *Nature communications*, 11(1):5575, 2020.

[9] Hanyu Gao, Thomas J Struble, Connor W Coley, Yuran Wang, William H Green, and Klavs F Jensen. Using machine learning to predict suitable conditions for organic reactions. *ACS central science*, 4(11):1465–1476, 2018.

[10] Connor W Coley, Wengong Jin, Luke Rogers, Timothy F Jamison, Tommi S Jaakkola, William H Green, Regina Barzilay, and Klavs F Jensen. A graph-convolutional neural network model for the prediction of chemical reactivity. *Chemical science*, 10(2):370–377, 2019.

[11] Zhengkai Tu and Connor W Coley. Permutation invariant graph-to-sequence model for template-free retrosynthesis and reaction prediction. *Journal of chemical information and modeling*, 62 (15):3503–3513, 2022.

[12] Daniel S Wigh, Joe Arrowsmith, Alexander Pomberger, Kobi C Felton, and Alexei A Lapkin. Orderly: data sets and benchmarks for chemical reaction data. *Journal of Chemical Information and Modeling*, 64(9):3790–3798, 2024.

[13] Philippe Schwaller, Alain C Vaucher, Teodoro Laino, and Jean-Louis Reymond. Prediction of chemical reaction yields using deep learning. *Machine learning: science and technology*, 2(1): 015016, 2021.

[14] Ryan-Rhys Griffiths, Leo Klarner, Henry Moss, Aditya Ravuri, Sang Truong, Yuanqi Du, Samuel Stanton, Gary Tom, Bojana Rankovic, Arian Jamasb, et al. GAUCHE: a library for gaussian processes in chemistry. *Advances in Neural Information Processing Systems*, 36: 76923–76946, 2023.

[15] Bojana Ranković, Ryan-Rhys Griffiths, Henry B Moss, and Philippe Schwaller. Bayesian optimisation for additive screening and yield improvements–beyond one-hot encoding. *Digital Discovery*, 3(4):654–666, 2024.

[16] Bojana Ranković and Philippe Schwaller. GOLLuM: Gaussian process optimized LLMs–reframing LLM finetuning through Bayesian optimization. *arXiv preprint arXiv:2504.06265*, 2025.

[17] Benjamin J Deadman. Wanted: Flow chemistry reaction data. *Chimia*, 79(6):390–395, 2025.

[18] Steven M Kearnes, Michael R Maser, Michael Wleklinski, Anton Kast, Abigail G Doyle, Spencer D Dreher, Joel M Hawkins, Klavs F Jensen, and Connor W Coley. The open reaction database. *Journal of the American Chemical Society*, 143(45):18820–18826, 2021.

[19] Daniel Mark Lowe. *Extraction of chemical structures and reactions from the literature*. PhD thesis, University of Cambridge, 2012.

[20] Damith Perera, Joseph W Tucker, Shalini Brahmbhatt, Christopher J Helal, Ashley Chong, William Farrell, Paul Richardson, and Neal W Sach. A platform for automated nanomole-scale reaction screening and micromole-scale synthesis in flow. *Science*, 359(6374):429–434, 2018.

[21] Derek T Ahneman, Jesús G Estrada, Shishi Lin, Spencer D Dreher, and Abigail G Doyle. Predicting reaction performance in c–n cross-coupling using machine learning. *Science*, 360 (6385):186–190, 2018.

[22] Cesar N Prieto Kullmer, Jacob A Kautzky, Shane W Krska, Timothy Nowak, Spencer D Dreher, and David WC MacMillan. Accelerating reaction generality and mechanistic insight through additive mapping. *Science*, 376(6592):532–539, 2022.

[23] Philippe Schwaller, Alain C Vaucher, Teodoro Laino, and Jean-Louis Reymond. Data augmentation strategies to improve reaction yield predictions and estimate uncertainty. *NeurIPS Workshop on Machine Learning for Molecules*, 2020.

[24] Thanh Nhat Nguyen, Thuy Tran Phuong Nhat, Ken Takimoto, Ashutosh Thakur, Shun Nishimura, Junya Ohyama, Itsuki Miyazato, Lauren Takahashi, Jun Fujima, Keisuke Takahashi, et al. High-throughput experimentation and catalyst informatics for oxidative coupling of methane. *Acs Catalysis*, 10(2):921–932, 2019.

[25] Mayk Caldas Ramos, Shane S Michtavy, Marc D Porosoff, and Andrew D White. Bayesian optimization of catalysts with in-context learning. *arXiv preprint arXiv:2304.05341*, 2023.

[26] Peter Sagmeister, Christine Schiller, Peter Weiss, Klara Silber, Sebastian Knoll, Martin Horn, Christopher A Hone, Jason D Williams, and C Oliver Kappe. Accelerating reaction modeling using dynamic flow experiments, part 1: design space exploration. *Reaction Chemistry & Engineering*, 8(11):2818–2825, 2023.

[27] Peter Sagmeister, Lukas Melnizky, Jason D Williams, and C Oliver Kappe. Simultaneous reaction- and analytical model building using dynamic flow experiments to accelerate process development. *Chemical Science*, 15(31):12523–12533, 2024.

[28] Klara Silber, Peter Sagmeister, Christine Schiller, Jason D Williams, Christopher A Hone, and C Oliver Kappe. Accelerating reaction modeling using dynamic flow experiments, part 2: development of a digital twin. *Reaction Chemistry & Engineering*, 8(11):2849–2855, 2023.

[29] Linden Robert McCabe Schrecker, Jose Pablo Folch, Klaus Hellgardt, King Kuok Hii, Joachim Dickhaut, Christian Holtze, and Andy Wieja. Solvent screening method. https://patentscope.wipo.int/search/en/detail.jsf?docId=WO2025073762, April 2025. WO Patent WO/2025/073762, PCT/EP2024/077742.

[30] Richard K Henderson, Concepción Jiménez-González, David JC Constable, Sarah R Alston, Graham GA Inglis, Gail Fisher, James Sherwood, Steve P Binks, and Alan D Curzons. Expanding gsk's solvent selection guide–embedding sustainability into solvent selection starting at medicinal chemistry. *Green Chemistry*, 13(4):854–862, 2011.

[31] Matthew B Plutschack, Bartholomäus Pieber, Kerry Gilmore, and Peter H Seeberger. The hitchhiker's guide to flow chemistry. *Chemical reviews*, 117(18):11796–11893, 2017.

[32] Ryan L Hartman, Jonathan P McMullen, and Klavs F Jensen. Deciding whether to go with the flow: evaluating the merits of flow reactors for synthesis. *Angewandte Chemie International Edition*, 50(33):7502–7519, 2011.

[33] Sarah L Boyall, Holly Clarke, Thomas Dixon, Robert WM Davidson, Kevin Leslie, Graeme Clemens, Frans L Muller, Adam D Clayton, Richard A Bourne, and Thomas W Chamberlain. Automated optimization of a multistep, multiphase continuous flow process for pharmaceutical synthesis. *ACS Sustainable Chemistry & Engineering*, 12(41):15125–15133, 2024.

[34] Jason D Williams, Peter Sagmeister, and C Oliver Kappe. Dynamic flow experiments for data-rich optimization. *Current Opinion in Green and Sustainable Chemistry*, page 100921, 2024.

[35] Linden Schrecker, Joachim Dickhaut, Christian Holtze, Philipp Staehle, Marcel Vranceanu, Klaus Hellgardt, and King Kuok Mimi Hii. Discovery of unexpectedly complex reaction pathways for the knorr pyrazole synthesis via transient flow. *Reaction Chemistry & Engineering*, 8(1):41–46, 2023.

[36] Linden Schrecker, Joachim Dickhaut, Christian Holtze, Philipp Staehle, Marcel Vranceanu, Andy Wieja, Klaus Hellgardt, and King Kuok Hii. A comparative study of transient flow rate steps and ramps for the efficient collection of kinetic data. *Reaction Chemistry & Engineering*, 9(5):1077–1086, 2024.

[37] Jason S Moore, Christopher D Smith, and Klavs F Jensen. Kinetics analysis and automated online screening of aminocarbonylation of aryl halides in flow. *Reaction Chemistry & Engineering*, 1(3):272–279, 2016.

[38] Jonathan P McMullen and Brian M Wyvratt. Automated optimization under dynamic flow conditions. *Reaction Chemistry & Engineering*, 8(1):137–151, 2023.

[39] Dawid Drelinkiewicz, Tom JA Corrie, and Richard J Whitby. Rapid investigation of the effect of binary and ternary solvent gradient mixtures on reaction outcomes using a continuous flow system. *Reaction Chemistry & Engineering*, 9(2):379–387, 2024.

[40] Louis J Diorazio, David RJ Hose, and Neil K Adlington. Toward a more holistic framework for solvent selection. *Organic Process Research & Development*, 20(4):760–773, 2016.

[41] Thomas M Dixon, Jeanine Williams, Maximilian Besenhard, Roger M Howard, James MacGregor, Philip Peach, Adam D Clayton, Nicholas J Warren, and Richard A Bourne. Operator-free HPLC automated method development guided by Bayesian optimization. *Digital Discovery*, 3 (8):1591–1601, 2024.

[42] Stefan Spange, Nadine Weiß, Caroline H Schmidt, and Katja Schreiter. Reappraisal of empirical solvent polarity scales for organic solvents. *Chemistry-Methods*, 1(1):42–60, 2021.

[43] Greg Landrum. RDKit: Open-source cheminformatics., 2006. URL https://www.rdkit.org.

[44] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010.

[45] Ryan-Rhys Griffiths, Jake L Greenfield, Aditya R Thawani, Arian R Jamasb, Henry B Moss, Anthony Bourached, Penelope Jones, William McCorkindale, Alexander A Aldrick, Matthew J Fuchter, et al. Data-driven discovery of molecular photoswitches with multioutput Gaussian processes. *Chemical Science*, 13(45):13541–13551, 2022.

[46] Nadine Schneider, Daniel M Lowe, Roger A Sayle, and Gregory A Landrum. Development of a novel fingerprint for chemical reactions and its application to large-scale reaction classification and similarity. *Journal of chemical information and modeling*, 55(1):39–53, 2015.

[47] Daniel Probst, Philippe Schwaller, and Jean-Louis Reymond. Reaction classification and yield prediction using the differential reaction fingerprint DRFP. *Digital discovery*, 1(2):91–97, 2022.

[48] Cynthia J. Burrows, Jason B. Harper, Wolfram Sander, and Dean J. Tantillo. Solvation effects in organic chemistry. *The Journal of Organic Chemistry*, 87(3):1599–1601, 2022. doi: 10.1021/acs.joc.1c03148. PMID: 35114791.

[49] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

[50] Peter I Frazier. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[51] Alexander Pomberger, AA Pedrina McCarthy, Ahmad Khan, Simon Sung, CJ Taylor, MJ Gaunt, Lucy Colwell, David Walz, and AA Lapkin. The effect of chemical representation on active machine learning towards closed-loop optimization. *Reaction Chemistry & Engineering*, 7(6):1368–1379, 2022.

[52] K.M. Borgwardt and H.P. Kriegel. Shortest-path kernels on graphs. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 8 pp.–, 2005. doi: 10.1109/ICDM.2005.132.

[53] Yilin Xie, Shiqiang Zhang, Jixiang Qing, Ruth Misener, and Calvin Tsay. BoGrape: Bayesian optimization over graphs with shortest-path encoded. *arXiv preprint arXiv:2503.05642*, 2025.

[54] Yilin Xie, Shiqiang Zhang, Jixiang Qing, Ruth Misener, and Calvin Tsay. Global optimization of graph acquisition functions for neural architecture search. *arXiv preprint arXiv:2505.23640*, 2025.

[55] Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. ChemBERTa: large-scale self-supervised pretraining for molecular property prediction. *NeurIPS Workshop on Machine Learning for Molecules*, 2020.

[56] Andres M Bran and Philippe Schwaller. Transformers and large language models for chemistry and drug discovery. In *Drug Development Supported by Informatics*, pages 143–163. Springer, 2024.

[57] Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and SM Ali Eslami. Conditional neural processes. In *International conference on machine learning*, pages 1704–1713. PMLR, 2018.

[58] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[59] Jixiang Qing, Becky D Langdon, Robert M Lee, Behrang Shafei, Mark van der Wilk, Calvin Tsay, and Ruth Misener. System-aware neural ODE processes for few-shot Bayesian optimization. *arXiv preprint arXiv:2406.02352*, 2024.

[60] Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32, 2019.

[61] Binxin Ru, Ahsan Alvi, Vu Nguyen, Michael A. Osborne, and Stephen Roberts. Bayesian optimisation over multiple continuous and categorical inputs. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8276–8285. PMLR, July 2020. URL `https://proceedings.mlr.press/v119/ru20a.html`.

[62] Edwin V Bonilla, Kian Chai, and Christopher Williams. Multi-task Gaussian process prediction. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL https://proceedings.neurips.cc/paper_files/paper/2007/file/66368270ffd51418ec58bd793f2d9b1b-Paper.pdf.

[63] Jasper Snoek, Kevin Swersky, Rich Zemel, and Ryan Adams. Input warping for Bayesian optimization of non-stationary functions. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1674–1682, Bejing, China, 22–24 Jun 2014. PMLR. URL https://proceedings.mlr.press/v32/snoek14.html.

[64] Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. In *Advances in Neural Information Processing Systems 33*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/f5b1b89d98b7286673128a5fb112cb9a-Abstract.html.

[65] Quoc V Le, Alex J Smola, and Stéphane Canu. Heteroscedastic Gaussian process regression. In *Proceedings of the 22nd International Conference on Machine learning*, pages 489–496, 2005.

[66] Mark Van der Wilk, Carl Edward Rasmussen, and James Hensman. Convolutional Gaussian processes. *Advances in neural information processing systems*, 30, 2017.

[67] Toby Boyne, Jose Pablo Folch, Robert M Lee, Behrang Shafei, and Ruth Misener. BARK: A fully Bayesian tree kernel for black-box optimization. *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

[68] Jose Pablo Folch, Shiqiang Zhang, Robert Lee, Behrang Shafei, David Walz, Calvin Tsay, Mark van der Wilk, and Ruth Misener. SnAKe: Bayesian optimization with pathwise exploration. *Advances in Neural Information Processing Systems*, 35:35226–35239, 2022.

[69] Mojmír Mutny, Tadeusz Janik, and Andreas Krause. Active exploration via experiment design in markov chains. In *International Conference on Artificial Intelligence and Statistics*, pages 7349–7374. PMLR, 2023.

[70] Jose Pablo Folch, Calvin Tsay, Robert Lee, Behrang Shafei, Weronika Ormaniec, Andreas Krause, Mark van der Wilk, Ruth Misener, and Mojmír Mutny. Transition constrained Bayesian optimization via Markov decision processes. *Advances in Neural Information Processing Systems*, 37:88194–88235, 2024.

[71] Andreas Krause and Carlos Guestrin. Nonmyopic active learning of Gaussian processes: an exploration-exploitation approach. In *Proceedings of the 24th international conference on Machine learning*, pages 449–456, 2007.

[72] Jonas Mockus. The application of Bayesian methods for seeking the extremum. *Towards global optimization*, 2:117, 1998.

[73] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

[74] Nery Riquelme, Christian Von Lücken, and Benjamin Baran. Performance metrics in multi-objective optimization. In *2015 Latin American computing conference (CLEI)*, pages 1–11. IEEE, 2015.

[75] Biswajit Paria, Kirthevasan Kandasamy, and Barnabás Póczos. A flexible framework for multi-objective bayesian optimization using random scalarizations. In *Uncertainty in Artificial Intelligence*, pages 766–776. PMLR, 2020.

[76] Carl Hvarfner, Erik Orm Hellsten, and Luigi Nardi. Vanilla Bayesian optimization performs great in high dimensions. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller,

Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 20793–20817. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/hvarfner24a.html.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Every claim in the abstract is repeated across the paper and justified, and expanded in the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See the conclusion.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: No theoretical results are in the paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: All details on the machine learning methods are provided in the appendix, furthermore the code to reproduce all results has been made public.

However, it is important to note that the exact laboratory set-up for the data creation will not be provided due to protected proprietary IP. Nonetheless, we are confident that similar results could be achieved in any flow chemistry set-up.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: This paper provides open access to the dataset, hosted on Kaggle. We also provide the code used to generate the experimental results.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: In Section 3.2, we briefly describe the experimental settings, and the models used. In the supplemental material, we describe in further detail these settings. We also provide the code, which includes these parameters.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: In the main section we do not provide error bars, since we base our results on full leave-one-out cross-validation. However, on the active learning section we report interquartile ranges on the performance of the algorithms across a variety of random initializations.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The supplemental material discusses the resources required to reproduce the experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms entirely to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: Potential positive societal impacts are discussed in the introduction and conclusion: more sustainable manufacturing. Potential negative ones are not discussed, as they are too broad and do not relate to this publication in particular.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The data that is released does not provide the risk for misuse, as the reaction described is not one with harmful uses.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper uses the existing Ethyl dataset, which is referenced in Section 1. We cite the creators, who are also authors on this paper. We provide the data alongside our new data on Kaggle, with the creators' permission.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release a dataset, which is hosted on Kaggle. We provide detailed description about the different parts of the dataset alongside the data itself. Every column in each table provides a description of the type and role of the data. The data was generated by the authors of the paper, who consent to using the asset.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: Since we use LLMs for their ability to represent molecules, we briefly introduce them, alongside specific pre-trained models. However, since LLMs do not make up a significant part of our experiments, we do not include much detail. Otherwise, LLMs were only used to help format tables, and make other formatting edits.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Details on the models and benchmarks

A.1 Benchmarking details

A.1.1 Regression

For regression on the dataset, we perform leave-one-out cross validation. For the single solvents, we leave out one solvent at a time. For the full data, we leave out one solvent ramp at a time. We measure the performance of the model on each leave-one-out data split, then take the mean of their performance across the dataset. We exclude any experiments involving acetonitrile and acetic acid, due to the observed side-reactions. In addition, when considering the testing in single solvent data, we create a set of single data-points by averaging over repeated measurements, in order to remove mean error weighting from the longer residence times, in order to understand if the models catch the time-series nature of the data.

A.1.2 Transfer learning

As above, we perform leave-one-out cross validation on the solvent ramps in the catechol dataset. However, when we train each model, we append the training data from the ethyl dataset, alongside a binary feature indicating which dataset each observation is from. We also replace the three outputs of the catechol dataset (SM, Product 2, Product 3) with a single column, Product, which is the sum of the two products. This allows us to compare across the two datasets, since the ethyl dataset only has a Product column.

A.1.3 Active learning and Bayesian optimization

For Bayesian optimization we optimize the weighted objective function:

$$f(S_A, S_b, b, \tau, T) = \lambda_1(P_2 + P_3) + \lambda_2 \frac{P_2}{P_2 + P_3} - \lambda_3 \frac{T - 175}{50} - \lambda_4 \tau \tag{4}$$

where S_A is solvent A, S_B is solvent B, b is the percentage composition of solvent B, τ is the residence time, T the temperature, and P_2 / P_3 the yields of Products 2 and 3 respectively. We set the weight parameter values to:

$$\lambda_1 = 5; \quad \lambda_2 = 1; \quad \lambda_3 = 3; \quad \lambda_4 = \frac{1}{20}$$

For the Upper Confidence Bound acquisition function we use the standard exploration parameter $\beta = 1.96$.

For locations with repeated measurements we simply consider average of all observations as the true product yields. All acquisition function optimizations are done through a simple exhaustive search of the space.

A.2 Model details

In this section, we provide the details necessary to reproduce the models used in the experiments. Any information that is not listed here can be found in our code, at `https://github.com/jpfolch/catechol_solvent_selection`.

A.2.1 Gaussian processes

We implement the GP models in this paper in BoTorch v0.13.0 [64]. We use the priors recommended by Hvarfner et al. [76], to ensure good performance across featurizations of different dimensions. We use an RBF kernel, with the lengthscale prior

$$p(\ell) = \mathcal{LN}(\sqrt{2} + \log\sqrt{D}, \sqrt{3})$$

All GPs were trained using the MLII likelihood (maximum a posterior), with a training timeout of 30 seconds. For all of the GP extensions (in Table 4), we use the Spange featurization.

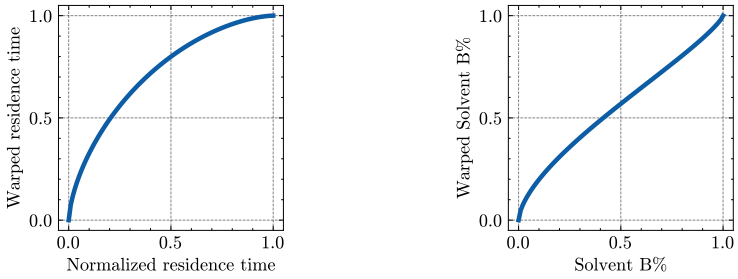


Figure 7: An example of a learned input warping, after training the GP on the full dataset.

BaselineGP. This model is a GP trained only using the residence time, and the temperature. This model does not factor in which solvent each experiment is from.

DeepGP. This model first trains a BaselineGP, then uses that as a mean function for another GP. In this way, far away from known solvents this model will fall back to the BaselineGP as a prior.

Decomposed kernel. We take inspiration from Ru et al. [61], and separate our kernel into two parts. Specifically, we consider the input to the model to be the concatenation of the solvent featurization, f, and the non-featurized inputs, x, which include residence time and temperature. We then use the following kernel,

$$k_{\text{decomp}}([x, f], [x', f']) = k_x(x, x') \cdot k_f(f, f') + k_x(x, x') + k_f(f, f')$$

Similarly to the deep GP, this allows the features in x to still contribute to the prediction, even when the unseen solvent is far from the known solvents.

Multitask GP. We use two different types of multitask GP in this paper. First, in Section 3.3, we use a multitask GP to represent each of the three measured yields. This kernel consists of a data kernel, and a task kernel,

$$k_{\text{MT}}([x, o], [x', o']) = k_x(x, x') \cdot k_o(o, o'),$$

where k_o is an $O \times O$ matrix (for this dataset, $O = 3$) that is used to learn the correlations between the outputs. Since all outputs are observed for each experiment, we can use a Kronecker structured kernel.

In Section 3.4, we use another multitask GP with 2 tasks, where each task corresponds to one of the two datasets. We use the same kernel as above, however only one task is observed at each reaction condition.

Input warping. In Section 3.3, we describe how the underlying chemistry is nonstationary. To attempt to address this, we take inspiration from Snoek et al. [63] and Balandat et al. [64], learning a bijective map $\phi : [0, 1] \to [0, 1]$ that can capture the nonlinear effect of mixing solvents. This map has hyperparameters that can be learned,

$$S_{A \cup B}(b) = (1 - \phi(b))S_A + \phi(b)S_B, \qquad \phi(b) = 1 - (1 - b^\alpha)^\beta,$$

where ϕ is the Kumaraswamy cumulative distribution function. We place a log normal prior on the parameters, $\alpha, \beta \sim \mathcal{LN}(0, \sqrt{0.3})$. This prior has median value of 1, which corresponds to a linear mapping.

We also use the input warping for the residence time. Since most of the reaction occurs in the first few minutes of the reaction, the lengthscale is far shorter compared to the later parts of the reaction. We find that this is indeed learned by the model, as shown in Figure 7; the mapping effectively 'spreads out' the observations early in the reaction, while compressing the later observations that tend to have a slower rate of change. Whilst the warping for the solvent composition learns a slight sigmoidal shape, we show experimentally in Section 3.3 that warping this feature does not improve regression performance.

A.2.2 Neural networks

Two types of neural network models were constructed for the regression tasks. The first was a standalone multilayer perceptron (MLP) model, and the second combined a large language model (LLM) backbone with an MLP head.

For the single-solvent task, the MLP model took as input the reaction time, temperature, and a feature vector representing the solvent. The network architecture consisted of two hidden layers with 128 and 64 neurons, respectively, each followed by ReLU activations and dropout (dropout rate of 0.5), and an output layer with 3 neurons.

For the mixed-solvent task, the MLP model used the same architecture, but the solvent input was computed as a sigmoid-weighted combination of the individual solvent feature vectors:

$$S_{A\cup B} = (1 - \sigma_\theta(b))\, S_A + \sigma_\theta(b) S_B,$$

where S_A and S_B are the featurizations of solvents A and B, b is the percentage of solvent B in the mixture, and σ_θ is a sigmoid function with trainable parameters θ.

The second model architecture used pretrained LLMs—**RXNFP** and **ChemBERTa**—to generate embeddings from reaction SMILES strings. For the single-solvent task, the SMILES representation of the reaction using the selected solvent was passed through the LLM to obtain the corresponding embedding. For the mixed-solvent task, the SMILES strings of the reactions carried out in solvents A and B, denoted RS_A and RS_B, were each processed independently through the LLM to produce embeddings $\mathbf{E}_A$ and $\mathbf{E}_B$, respectively. These embeddings were then combined using a sigmoid-weighted sum:

$$\mathbf{E}_{A\cup B} = (1 - \sigma_\theta(b))\, \mathbf{E}_A + \sigma_\theta(b) \mathbf{E}_B,$$

where b is the percentage of solvent B in the mixture and σ_θ is a sigmoid function with trainable parameters θ.

The resulting embedding was concatenated with the time and temperature, and passed through an MLP with the same architecture as the standalone MLP model. The LLM backbones were kept frozen during training, and only the MLP head was optimized.

The `ChemBERTa` model and tokenizer used were `seyonec/ChemBERTa-zinc-base-v1`, loaded via the Hugging Face `transformers` library. Similarly, the pretrained `RXNFP` model and tokenizer used are available from the `rxnfp` repository.

All models were trained using a learning rate of 0.001, a batch size of 32, for up to 400 epochs, or until reaching a maximum runtime of 720 minutes.

A.2.3 ODE

The ODE models were trained with a learning rate of 0.001, and 100 epochs. For the latent state and latent dynamics, we used a 32-dimensional space, and for all of the other representations we used a 64-dimensional space. Further information can be found in the provided code.

A.3 Additional results

We showcase additional results for Neural Processes [57] and graph Gaussian processes [52, 53] in table 7.

B Details on data collection

B.1 Reactor details

Here we include the reactor and detail procedures.

The automated reactor setup used to collect the data is shown in Figure 8. Knauer Azure 4.1S pumps fitted with stainless steel 10 mL pump heads were used as pumps 1 and 2. All tubing used for the entire reactor was made of 316 stainless steel (1.5875 mm OD, 1 mm ID). An Agilent inline jet weaver HPLC mixer (350 μL volume) was used as an inline mixer to ensure the reactant solution was homogeneous before entering the reactor. An Agilent 6890 GC oven was used to heat the stainless steel coiled reactor (1.5875 mm OD, 1 mm ID, 7.95 mL volume) during the reaction to the desired temperature. A customized cooling system made from an aluminum block and a Peltier assembly

Table 7: Regression performance on the single solvent dataset. Mean squared error (MSE) and negative log predictive density (NLPD) are averaged across all leave-one-out data splits. We include the shortest path kernel (sp) and the exponential shortest path kernel (esp).

		Single solvent	
Model	Featurization	MSE (↓)	NLPD (↓)
NP	acs	0.153	-1.173
	drfps	0.139	-1.587
	fragprints	0.135	-1.495
	spange	0.089	-1.472
GraphGP	sp	0.046	2.464
	esp	1.068	2.453

was then placed inline to rapidly cool the flow of solution and quench the reaction. A Vici four port-2 position sampling valve followed the Peltier to sample small aliquots (500 nL) into the HPLC for online analysis measurements of the reaction. An IDEX 1000 PSI BPR was then placed before the waste tubing of the reactor to depressurize the reaction solution back to atmospheric pressure. The pumps, oven and Vici valve were automated by code developed in house in Python.

B.1.1 Methods

A typical reaction run was performed as following:

1. The reactant solutions were made up by adding allyl phenyl ether (50 μL) and the internal standard - ethyl benzene (50 μL) in to both solvent A and solvent B (250 mL) in separate volumetric flasks.

2. The reactant pumps were primed with their respective solvents and pumped through the system at 1 mL min^{-1} for 15 minutes.

3. The pumps were then primed with the reactant solutions and pumped through the system at 1 mL min^{-1} for 5 minutes.

4. The HPLC was started and a sequence was created to record external sampling via the Vici Valve.

5. The python code that runs the experiments was then initialized and the experiment was started.

6. Once the reaction run was completed, the reactor is flushed with their respective solvents for 10 minutes at 1 mL min^{-1}, followed by a flush of the system with a miscible solvent (usually IPA) and cleaned for the next reaction. The data was stored in a SQL database and is then deconvolved offline.

All the data-points recorded were reported in the dataset, and the only outliers that were removed were those slugs that experienced a step-up in flow-rate while in the reactor, as this has been shown to add bias to the data [36].

B.2 Fine-tuning calibration via optimization

The HPLC data we obtained is uncalibrated, which means we cannot calculate yields directly from the peak areas collected from online HPLC measurements. However, the yields of each product follows the linear relationship with peak areas:

$$y_{product} = \epsilon_{product} \times \frac{c_{IS}}{c_0} \times \text{peak\_ratio} \tag{5}$$

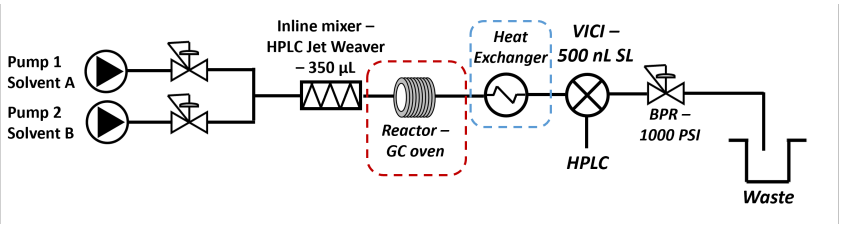


Figure 8: Piping & instrumentation diagram of the automated continuous flow coiled reactor used to collect the transient flow data reported in this paper.

where c_{IS} is the internal standard concentration in mol L^{-1}, c_0 is the initial concentration of starting material in mol L^{-1}, and ϵ is the *calibration constant*. The peak_ratio refers to value given by dividing the area of the peak of interest (starting material, product 2 or product 3) by the peak area of the internal standard. This constant is calculated by performing calibrations of the HPLC detector with injections of pure compounds at different concentrations, while keeping the internal standard concentration constant, and therefore observing the linear relationship and obtaining the response factor of the compounds. Obtaining a pure sample of Product 2 and Product 3 however, turned out to be particularly difficult due to the compounds being isomers, making the separation of the pure products tough. Therefore, we instead focused on using the estimates we had and then fine-tuning them via an optimization procedure.

Our initial HPLC tests gave us the following estimates:

$$\hat{\epsilon}_{SM} = \frac{1}{1.5}; \quad \hat{\epsilon}_{P2} = \frac{1}{3}; \quad \hat{\epsilon}_{P3} = \frac{1}{3}$$

From here, we decided to fine-tune the estimates in order for the calculated yields to ensure the reaction yields were mass balanced. We identified specific measurements where we expected full conversion (i.e. the sum of yields should be 100), and we further allowed for experimental concentrations to vary according to the error in the laboratory analytical pipettes used for making the reactant solutions. This results in the following optimization problem, where we penalized deviation from our initial calibration measurements, and deviation from full conversion at specified measurements $\mathcal{K}$:

$$\begin{aligned}
\min_{\{c_i, \epsilon_j\}} \quad & \alpha \sum_i (c_i - 2.25)^2 + \beta \sum_j (\epsilon_j - \hat{\epsilon}_j)^2 + \gamma \sum_{k \in \mathcal{K}} \left(\sum_j y_{kj} - 100 \right)^2 \\
\text{where} \quad & y_{ij} = \text{const} \cdot \text{peak_ratio}_{ij} \cdot \epsilon_j \cdot c_i, \quad \forall i = 1, ..., 1227; j \in \{SM, P2, P3\} \\
& c_i = c_{i'} \qquad \text{if } i, i' \text{ are in the same experimental run}
\end{aligned}$$

with constraints to restrict total yield under 100% and possible errors in concentrations:

$$\begin{aligned}
\sum_j y_{ij} &\leq 100, \quad \forall i \\
c_i &\in [1.25, 2.5], \quad \forall i \\
0.2 &\leq \epsilon_j \leq 0.5, \quad \forall j
\end{aligned}$$

where:

- c_i are the corrected concentration ratios,
- ϵ_j are the calibration scaling factors for each compound,
- peak_ratio_{ij} are the observed HPLC peak area ratios,
- $\mathcal{K}$ is the set of indices where full conversion is expected,
- α, β, and γ are weighting parameters.

we optimized with $\alpha = \beta = \gamma = 1$, optimized using `scipy`'s `minimize` function with the Sequential Least Squares Programming (SLSQP) algorithm. To select the initial values, we used a 100,000 initial grid search. This resulted in the following parameter estimates:

$$\epsilon_{SM} = 0.525; \quad \epsilon_{P2} = 0.222; \quad \epsilon_{P3} = 0.361$$

B.3 Spange descriptor interpolation

The descriptors from Spange et al. [42] were obtained from the supplementary material on the paper. However, there are a few values missing from some rows, including for the solvents we gathered data for. In order to estimate the missing values, we trained a multi-task Gaussian process model on the whole table, under a Taniamoto kernel, which we then used to predict the missing values that are used for all the main methods in the paper.