Automated pH Adjustment Driven by Robotic Workflows and Active Machine Learning

A. Pomberger,[1,2] N. Jose,[1,3,4] D. Walz,[5] J. Meissner,[5] C. Holze,[5] M. Kopczynski,[5] P. Müller-Bischof[6] and A. A. Lapkin[1,2,3]

[1]Department of Chemical Engineering and Biotechnology, University of Cambridge, CB3 0AS Cambridge, United Kingdom
[2]Innovation Centre in Digital Molecular Technologies (iDMT), Yusuf Hamied Department of Chemistry, University of Cambridge, CB2 1EW Cambridge, United Kingdom
[3] Cambridge Centre for Advanced Research and Education in Singapore, CARES Ltd. 1 CREATE Way, CREATE Tower, 138602 Singapore
[4] Accelerated Materials Ltd, 71-75, Shelton Street, WC2H 9JK London, United Kingdom
[5] BASF SE, Carl-Bosch-Strasse 38, 67056 Ludwigshafen am Rhein, Germany
[6] Vienna University of Technology, Karlsplatz 13, 1040 Vienna, Austria

Authors contributions

A. Pomberger developed the idea in discussions with D. Walz, J. Meissner, C. Holze and M. Kopczynski. A. Pomberger wrote the code and conducted the ML screening together with P. Müller. N. Jose developed and configurated the control code for the robot. AAL developed the project concept, secured funding and supervised the project. All authors discussed the results and contributed to the final manuscript.

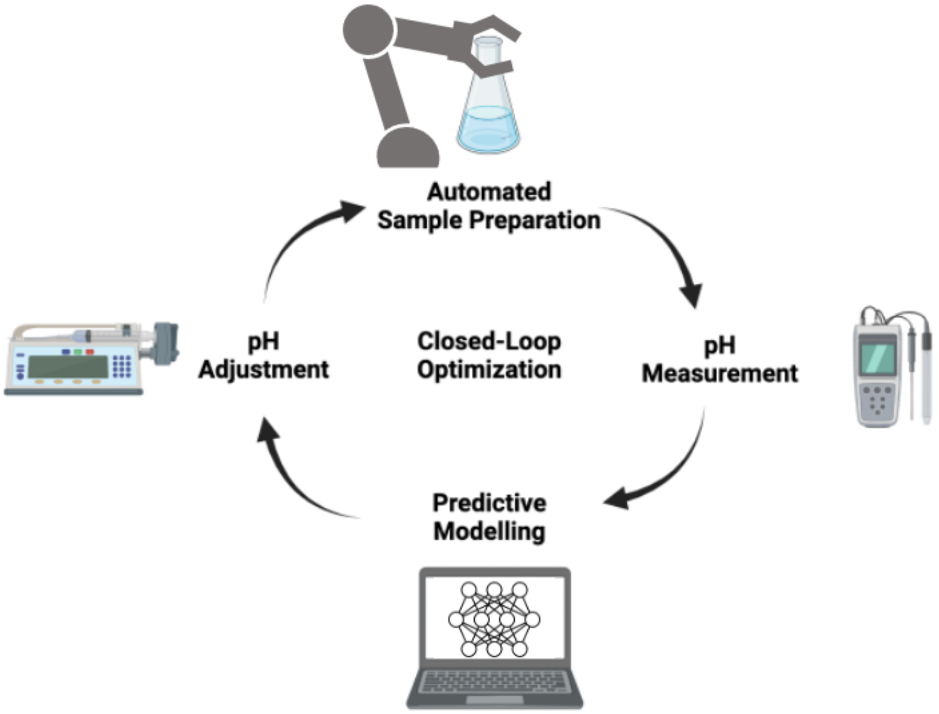
Abstract

Buffer solutions have tremendous importance in biological systems and in formulated products. Whilst the pH response upon acid/base addition to a mixture containing a single buffer can be described by the Henderson-Hasselbalch equation, modelling the pH response for multi-buffered poly-protic systems after acid/base addition, a common task in all chemical laboratories and many industrial plants, is a challenge. Combining predictive modelling and experimental pH adjustment, we present an active machine learning (ML)-driven closed-loop optimization strategy for automating small scale batch pH adjustment relevant for complex samples (e.g., formulated products in the chemical industry). Several ML models were compared on a generated dataset of binary-buffered poly-protic systems and it was found that Gaussian processes (GP) served as the best performing models. Moreover, the implementation of transfer learning into the optimization protocol proved to be a successful strategy in making the process even more efficient. Finally, practical usability of the developed algorithm was demonstrated experimentally with a liquid handling robot where the pH of different buffered systems was adjusted, offering a versatile and efficient strategy for a pH adjustment processes.

Corresponding author: A. Lapkin, email: aal35@cam.ac.uk

***Keywords**: pH adjustment; active machine learning; buffered systems; closed-loop optimization*

Graphical Abstract



Introduction

Adjusting pH is an important step in the production of cosmetic formulations, liquid detergents, treatment of industrial wastewater or within biopharmaceutical drug manufacturing.[1-6] The process itself is often very time intensive due to complex proton partitioning equilibria and represents a challenging control problem resulting from the intrinsic non-linearity of the pH value.[7] Additionally, buffer chemicals, weak acids or bases that can donate or accept protons, often used in the formulations to maintaining the pH within a narrow margin upon acid/base addition, complicate the process of pH adjustment. While single buffered systems can be described using the Henderson-Hasselbalch equation, developing models for multiple poly-protic buffers (e.g., phosphate and citrate) remains an ongoing challenge.[8, 9]

Commercial and literature-reported pH adjustment strategies are typically either based on proportional-integral-derivate (PID) control or model predictive control (MPC); both come with limitations. The PID control strategy continuously calculates deviation of the measured value from the target value, and applies a correction based on a proportional, integral or derivative correction strategy.[10, 11] A typical example are bioreactors which often operate in fed-batch mode, relying on PID pH control to allow for the maintenance of specific conditions required by the biological cultures.[12-15] While no chemical information except the continuous measured pH is needed for PID, a loss of information needs to be accepted since insights into the chemical system cannot be implemented into future pH adjustments, as opposed to model based strategies. More recent pH adjustment approaches are based on MPC arrays, e.g., Altinten *et al.* describe a generalized predictive control for continuous flow pH adjustment[16], Helmy *et al.* relied on multi-linear regression[17] and Alkamil *et al.* used a fuzzy artificial neural network (ANN) outperforming a PID-control.[18] Others have also expanded on using ML based MPC strategies for this same purpose.[19, 20] A major challenge associated to MPC-based pH adjustment is operating in a low data regime, particularly of interest for high-throughput small-scale pH adjustment, as opposed to continuous pH adjustment. Aside from automated strategies, pH adjustment process is also often conducted manually in a R&D stage which is time consuming, requiring approximately five to seven minutes per sample (Table 1: entries 7, 10, 13).

The modern digitalization of research facilities allows the relatively fast and easy accumulation of experimental data, which can be used to accelerate subsequent workflows by employing transfer learning (TL). TL represents the method of pretraining ML models for one task and subsequentially using the trained model for a similar prediction task.[21] Typically, the auxiliary model would profit from an availability of large pretraining datasets, whereas the target model is fine-tuned on a smaller dataset.[22-25] Process chemists often modify the composition of formulated products (e.g., liquid laundry detergents) to fine tune product properties (e.g., viscosity). While small modifications do not change the overall composition greatly, the pH response (titration curve) does change and thus the sample requires a new titration strategy every time.

Herein, we aim to employ a data-driven strategy for pH adjustment, benefitting from active learning and robotic facilities for experimental evaluation. We compare different surrogate models, machine-readable data representations and initialization strategies for the development of an active ML-based pH adjustment strategy of multi-buffered poly-protic mixtures. By employing TL, benefiting from previously generated data, we aim to demonstrate this novel strategy for pH adjustment and keep the process efficient, even under an extreme low data regime.

Our approach for active ML-driven closed loop optimization is shown in Figure 1a. A chosen ML model is initially trained within a low data regime (here three datapoints) and used for predicting the unknown (ground truth) full titration curve (Figure 1b). Subsequently, conditions towards the target pH are selected using a custom acquisition function. In active learning and Bayesian optimization the acquisition function is typically a trade-off between exploration to reduce model uncertainty and exploitation towards the target value. For the pH adjustment we choose a purely exploitative approach by selecting the minimizer of the difference between the model predictions and the target pH as the next experimental condition (Figure 1c). This is possible as the pH curve is monotonous, hence the algorithm will converge to the target pH. Until the target pH has been reached, the dataset is continuously updated and the model is retrained for the next iteration. We coupled our active ML-guided pH adjustment approach with a liquid handling robot and successfully adjusted a set of chemically different binary buffered mixtures. After a comparison of different surrogate models, we identified Gaussian processes (GP) as the best performing model. Moreover, we managed to boost efficiency of the process by utilizing TL strategies, thus decreasing the required iterations of pH adjustment.

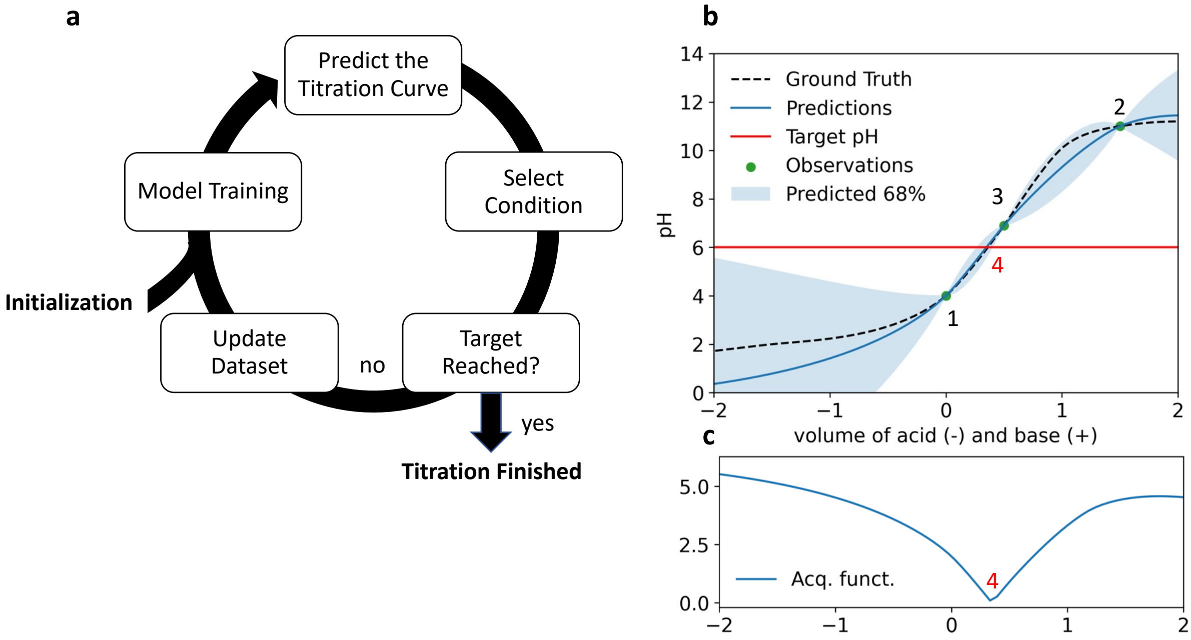


Figure 1. An overview of the ML-driven pH adjustment strategy (a) Design of the closed-loop optimization toward pH adjustment (b) Illustration of ML model predictions and decision making using the acquisition function, see minimum at 4. (c). The numbers represent the order of the observations, and the red font color represents the datapoint to be acquired in the next iteration. Both acid and base volume addition are represented on the x axis, where the negative values account for acid volume and the positive values account for base volumes.

Materials and Methods

Active ML-driven Closed-loop Optimization

A chosen surrogate model is trained to map the input data (features) to the correlated output data (labels) - using a model-dependent data-processing architecture - by iteratively optimizing the model, i.e., minimizing the loss for all fitted datapoints.[26-28] The predictive model performance is subsequentially evaluated on a held-out test dataset – the model predictions are compared to the true values and the deviation is typically quantified via the residuals metric of root mean squared error (RMSE). By applying this strategy in an iterative manner, it can be used to navigate through the parameter space, efficiently searching for desired conditions.[29-31] In the case of pH adjustment this refers to the amount of acid or base to achieve a target pH. Algorithm 1 illustrates the control code used for automated pH adjustment.

```
Set parameter space (min/max volumes of acid/base)
Set $pH_{target}$
Randomly select 3 conditions
Execute the experiments and generate initial dataset

While $(pH_{target} – pH_{actual}) > 0.2$:
    Train the model on dataset
    Predict pH for undiscovered parameter space
    Identify conditions for the predictions that is closest to $pH_{target}$
    Conduct the selected experiment
    Update the training data
```

Algorithm 1. Closed-loop optimization.

To assess the performance of different surrogate models, particularly within a low data regime, and to deliver promising predictions, we conducted a comparative study between several models. Four commonly used ML models were chosen to understand their respective benefits and limitations: linear regression, random forest (RF),[32] Gaussian process (GP)[33] and artificial neural networks (ANN).[34] Hyperparameters for each model were optimized *a priori*, see SI for more detail (Section 3).

Robotic Platform

Based on the need to generate training data, as well as to demonstrate the active ML-based closed-loop pH adjustment process, we developed a robotic platform capable of mixing buffer solutions, measuring pH value and automatically conducting pH adjustment (Figure 2). Here, the X/Y/Z labels refer to buffer stock solution that can be pumped into glass vials (24 x 15 mL) positioned on the robotic wheel, acting as an auto sampler. On subsequent positions of the wheel, pH measurement and addition of acid/base can be conducted. After each pH adjustment process, the electrode is cleaned with deionized (DI) water to avoid cross-contamination between the samples. Technical design of the bespoke robotic platform was based on previous studies.[35, 36] We utilized FLab, a Python-based library, for facilitating communication between the motors, pumps, the pH electrode and the implementation of the ML based optimization algorithm.[37] See SI (Figure S1) for more detailed images and information on the robotic platform.

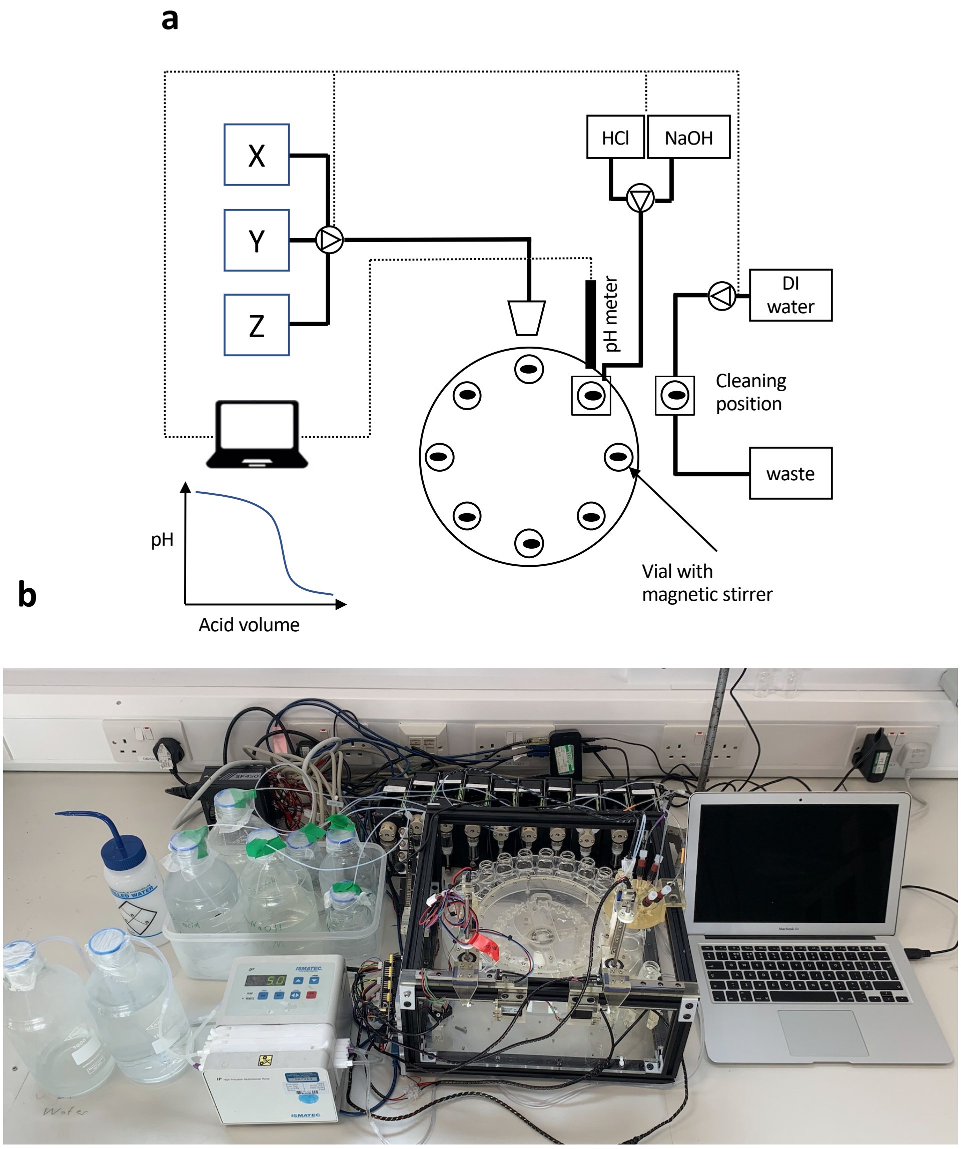


Figure 2. Schematic (a) and image (b) of the robotic pH adjustment platform. X/Y/Z indicate the stock solutions of buffer chemicals. For simplification not all 24 vials have been drawn on the robotic wheel. See SI for detailed labelled explanation of all components.

Results and Discussion

Closed-loop Optimization pH-Adjustment

The performance of pH adjustment can vary significantly, depending on the complexity of the system response towards the addition of a titrating agent. To demonstrate the broad applicability of the pH adjustment strategy, we tested our approach on a variety of different chemical systems. 18 experimentally generated datasets of binary buffered mixtures, containing the acid/base volume addition as the input and the measured pH value as the output were used, see Table 1. Based on the existence of this experimental data, simulated closed-loop optimization was conducted. The majority of the datapoints were held out and only a randomly selected batch of datapoints for initializing the model was used. The strategy (Figure 1) was applied, and the target pH was set to pH 6, with an acceptable deviation of a pH value of ± 0.2. For mixture 6 the initial pH of the sample was already within the target pH margin so the objective was set to pH 8. This workflow was conducted 10 times for each dataset and the mean/standard deviation of the number of iterations needed to achieve the target pH were calculated.

Table 1. A list of buffers used in this study.

Index	Buffer 1	Buffer 2	Ratio
1	Acetate	Citrate	1:1
2	Acetate	Citrate	1:2
3	Acetate	Citrate	2:1
4	Acetate	KH_2PO_4	1:1
5	Acetate	KH_2PO_4	1:2
6	Acetate	KH_2PO_4	2:1
7	Ammonium	Acetate	1:1
8	Ammonium	Acetate	1:2
9	Ammonium	Acetate	2:1
10	Ammonium	Citrate	1:1
11	Ammonium	Citrate	1:2
12	Ammonium	Citrate	2:1
13	Ammonium	KH_2PO_4	1:1
14	Ammonium	KH_2PO_4	1:2
15	Ammonium	KH_2PO_4	2:1
16	Citrate	KH_2PO_4	1:1
17	Citrate	KH_2PO_4	1:2
18	Citrate	KH_2PO_4	2:1

Featurization included the concentration of both buffer chemicals, the volume of acid/base as well as chemical information such as pKa values, the number of protons per buffer and the initial pH value. Figure 3a illustrates the varying prediction performance of four chosen surrogate models, using four observations for training. By comparing the single surrogate model predictions against the ground truth it becomes visible that linear regression delivered the worst fit, as expected, whereas GP delivered the best performance. Moreover, the characteristic piece-wise constant predictions, arising from the decision-tree based model architecture of the RF are visible, see Loh.[38] Figure 3b illustrates the optimization trajectory of ANN (buffer system 2) towards the target pH 6, i.e. how the model conducts sampling of experimental datapoints to find the target pH 6. It is visible that the algorithm initially requires approximately two iterations to explore the response and then starts to exploit towards the objective.

A broad comparison of the 18 buffer systems and the choice of the ML model was conducted to assess the required number of iterations to conduct pH adjustment (Figure 3c). Chemical systems with multiple protons tend to have more linear areas whereas a system comprised of two single chemicals (e.g., ammonium, acetate) tends to be less smooth, see SI Figure S2. In addition to the number of protons, the pKa values are also important as they indicate the location of the inflection point that will likely influence the system response of the binary mixture. Systems containing two poly-protic buffers (e.g. citrate and phosphate) tend to require fewer iterations compared to systems containing monoprotic buffers such as ammonium or acetate. Given the variety of the tested buffer chemicals, we believe that a wide variety of buffer systems can be represented using the dataset, i.e. the strategy should be applicable to samples containing other pH-sensitive chemicals which are not directly represented in this study.

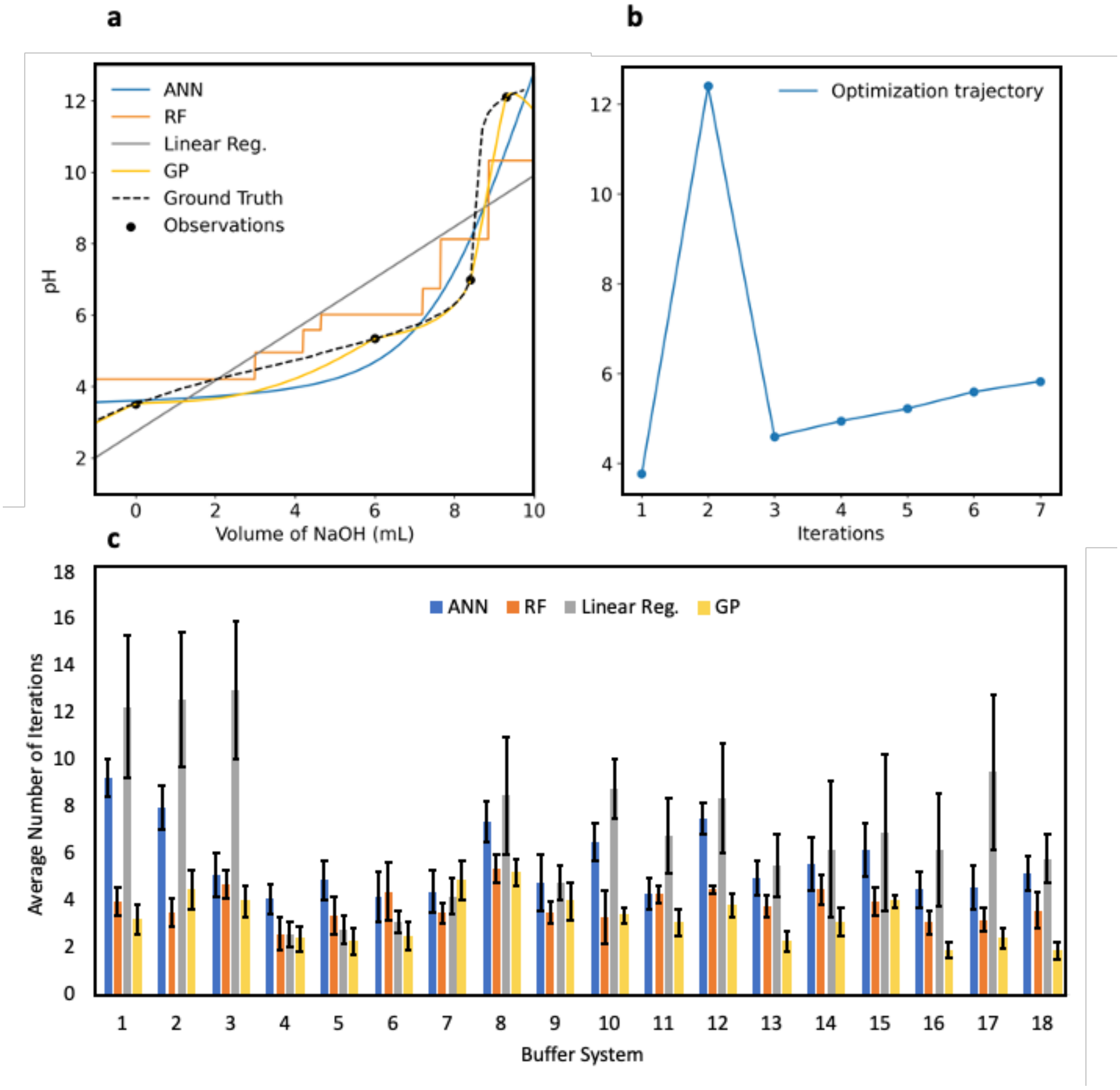


Figure 3. Illustration of the active ML pH adjustment (a) Insights into prediction performance of four models after four observations (buffer system 1) and comparison to the ground truth. (b) Optimization trajectory towards the target pH 6 (buffer system 2) using a ANN surrogate model (c) Comparison of the required iterations to reach target pH using four different ML models for 18 different binary buffered systems. The features include the pKa values as well as the initial pH values. The error bars represent the error on mean value. See Table 1 for indexed buffer system positions.

Linear regression clearly seems not to fit the datapoints well due to non-linear pH response, but it was conducted as a reference. It requires a high number of iterations for systems containing many polyprotic components, e.g., citrate, see Figure 3a,b. Overall, most of the systems could be adjusted within 3-4 iterations using three datapoints to initialize the optimization, thus giving a total of 6-7 required steps. On average, RF required 3.4 ± 0.3 iterations, ANN required 5.6 ± 1.0 iterations and GP required 3.1 ± 0.6. Here and in the following the reported values refer to the mean and the error of the mean value of 10 single iterations, see SI Eqn. S2. Our analysis shows that using the GP model gives the best results with the lowest number of iterations within the optimization loop.

Featurization Effects

Representing chemical compounds in a machine-readable format is considered a challenge in chemoinformatics due to its effect on different surrogate models and, thus, their predictive performance.[39] Previous literature has led to ambiguous outcomes on whether the addition of chemical information within low data regimes, such as the initialization of active-ML search strategies, is beneficial.[31, 40] To learn more about featurization effects on this specific application, we compared two input feature sets. The large feature set contains information on the components' concentrations, component pKa values, number of protons a buffer can accept/donate and the initial pH value of the buffer mixture (prior to any acid/base addition). The small feature set contains only information on the components' concentrations but no chemical insights.

As one can see in Figure 4, the performance of different informative features only minimally varies across the set of 18 systems. On average, using the large feature set resulted in 3.1 ± 0.6 iterations and the small feature set in 3.2 ± 0.6. While the results of the GP performance without chemical information might seem surprising, it must be noted that additional features increase the number of model parameters that need to be learned, as shown in other previously reported active ML studies by Pomberger *et al.*[40] The model initialization for each single system was conducted with 5% of the training data instead of a consistent number of datapoints. While the number of initialization datapoints varies, the focus is on the relative comparison of the different feature sets.

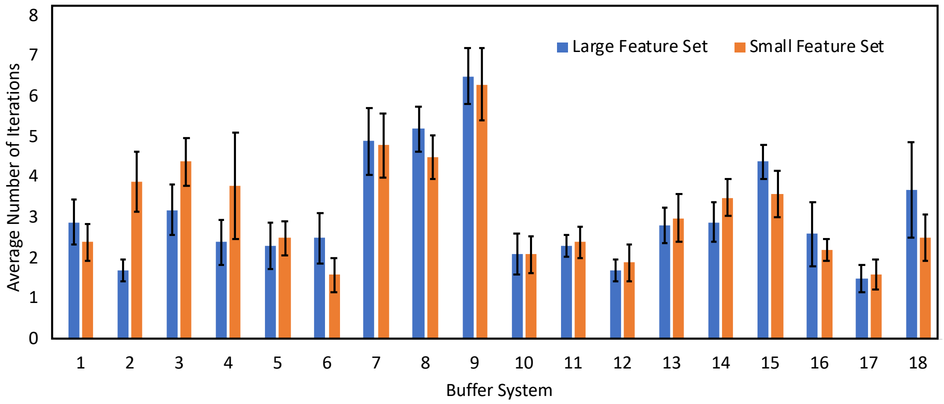


Figure 4. Illustration of the required active learning iterations using GP to reach target pH for a set of buffer systems using two feature sets. Error bars represent the error on the mean value. See Table 1 for information on the indexed buffer system positions.

As a result of the very similar outcome of the experiments (addition or exclusion of chemical information) it can be assumed that the strategy can be applied to chemical systems in a generic manner, specifically without exact knowledge of the chemical composition or chemical structure – a challenge faced when e.g., working with confidential industrial data. Due to the slightly better performance, all further experiments were conducted using the large feature set within this study.

Variation of the Number of Datapoints for Model Initialization

The choice of the number of datapoints (obtained via random selection) for initializing the closed-loop cycle impacts the preliminary surrogate model's prediction performance. While more initial datapoints could be considered as advantageous to train more accurate surrogate models, using fewer datapoints accelerates the overall adjustment process and might allow to selectively choose the subsequent datapoints based on the model's prediction instead of initial random allocation. Within this case study we aim to identify this effect by comparing a GP, initialized with two, three and four random datapoints.

We investigated different sized initialization datasets for all 18 binary buffer systems, see Figure 5. When analyzing the results, we want to directly compare the total number of datapoints (i.e. pH measurements) required to obtain the target pH, hence the sum of the number of datapoints within the initialization dataset and the number of datapoints obtained during the experimental iterations. Overall, using only two initial datapoints resulted in the fastest method, requiring on average 5.8 ± 0.6 total pH measurements, followed by 6.3 ± 0.6 and 6.8 ± 0.5 pH measurements for three and four initial datapoints, respectively. When initializing a model with two datapoints, the subsequent two datapoints are chosen selectively as opposed to using four random datapoints for initialization. The results indicate that the selective choice of the active ML strategy seems to be beneficial over random datapoint allocation, irrespective of the fact that the preliminary model is solely trained on two datapoints.

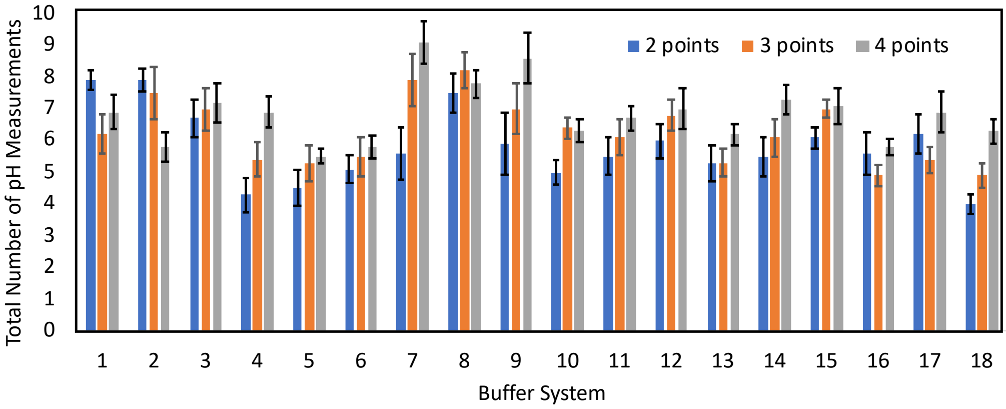


Figure 5. Illustration of the effect of variation in the number of initialization datapoints on the total number of pH measurements necessary, using the large feature set and GP model. The deviation represents the calculated error on the mean value. See Table 1 for information on the indexed buffer system positions.

Transfer Learning-Accelerated Closed-Loop Optimization

Harvesting existing data to facilitate knowledge transfer was explored, to measure if preliminary models have a better understanding of the system response to acid/base additions, thereby accelerating the process of pH adjustment. In detail, we investigated whether prior knowledge of the pH response of single components may accelerate closed-loop pH adjustment of binary buffered mixtures. For example, information on pH response of ammonium and acetate was provided when conducting the pH adjustment of an ammonium-acetate sample. The titration information of the pure single-component buffer chemicals was combined with the initialization data and used for training the initial model. As shown in Figure 6, the observable trend is that the addition of prior information improves the optimization performance.

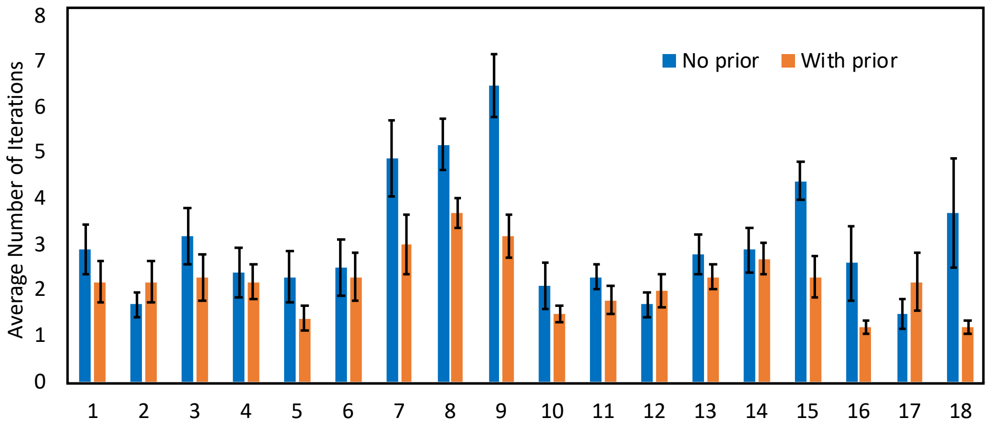


Figure 6. Comparison of active ML-driven pH adjustment using GP and the full feature set with and without the implementation of prior information. The error bars represent the error on the mean value. Model initialization was conducted with 5% of the training data. See Table 1 for information on the indexed buffer system positions.

Overall, using the GP alone without any prior information (just the initialization data) required 3.1 ± 0.6 iteration cycles, whereas, when implementing prior information of the single components, the number of iterations could be decreased down to 2.2 ± 0.4. Particularly challenging chemical systems, such as ammonium-acetate could be adjusted in significantly fewer number of iterations.

Real-Time Automated pH Adjustment

After developing a strategy for automating the experimental workflow via a robotic platform along with an algorithmic strategy for controlling the addition of acid/base separately, we then aimed to merge both efforts. Using Flab, the control code of the liquid handling robot allows direct interaction with the algorithmic pH adjustment strategy – the measured data is directly used for ML surrogate model training. The results of the subsequent decision making (next conditions to evaluate experimentally) is passed to the liquid handling robot. The adjustment process and decision making can be monitored in real-time, as shown in Figure 1b.

While previous experiments were initiated with two - four randomly selected datapoints, we now initiated the pH adjustment process with a single datapoint aiming to decrease the overall number of required experimental observations. After initial pH measurement (volume of added acid/base = 0) the selected volume of titrant is added, and data acquisition commences. Figure 7 illustrates the results of the automated pH adjustment, representing the average of three single experimental evaluations. The plot indicates the clear differences between various buffered systems, ranging from two iterations (citrate-phosphate) to eight iterations (acetate-citrate). To demonstrate the performance of our approach for a chemically extremely complex equilibrium system and the feasibility of the GP to model the data we conducted successful pH adjustment of a sample containing up to four buffer chemicals. For a mixture of citrate, phosphate, ammonium and acetate the target pH 6 was achieved within 3.7 ± 0.4 iterations, thus demonstrating the versatility of the presented data-driven strategy. Overall, 4.7 ± 0.4 iterations were required to adjust the sample mixtures to the target pH 6.

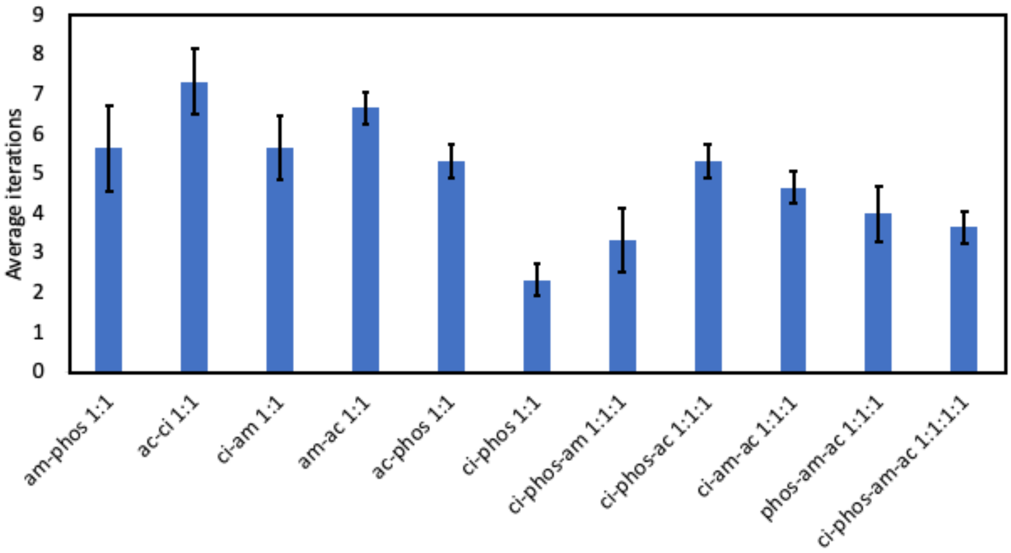


Figure 7. Results of the experimental case study using the robotic platform and the developed active ML closed-loop algorithm to conduct automated pH adjustment of unknown buffered systems. For phos-am-ac 1:1:1 the target pH was set to 8 since the initial sample already yielded approximately a pH of 6. The error bars represent the error on the mean value, see SI Eqn. S2. Abbreviations: am: ammonium, phos: KH_2PO_4, ac: acetate, ci: citrate.

Conclusions

Within this study, we present a method to adjust the pH of several multi-buffered polyprotic solutions to aid chemical laboratories dealing for formulation chemistry. A set target pH can be achieved via an iterative workflow in a fully automated manner, using a robotic platform informed by an active machine learning-based optimization strategy.

Specifically, a Gaussian process was used to predict the titration curves of several mixtures and guide the pH adjustment towards a set target pH. Chemical inputs were featurized containing increasing levels of chemical information, delivering only marginally better efficiency. This can be regarded as advantageous since it allows to implement this approach for systems without the requirement of molecular information, particularly beneficial when dealing with confidential industrial formulation samples or when the composition of the sample has not yet been characterized in detail. Applying transfer learning to the optimization cycle significantly boosted the performance, thus highlighting the main advantage of ML-driven pH adjustment over PID controlled or manual pH adjustment. Since it is common for samples in high-throughput formulation preparation to differ in only one or a few parameters in their compositions, learning from previous pH adjustments and transferring the obtained system knowledge into a new pH adjustment process has been quantitatively shown to benefit the overall workflow.

In an attempt to balance the number of initially randomly chosen datapoints to selectively chosen datapoints it was observed that the overall sample efficiency improved when using less initial data points. Finally, the strategy was demonstrated within a real experimental study with chemical systems containing up to four buffers – connecting the optimization algorithm and a robotic platform for conducting sample preparation and fully autonomous pH adjustment.

The developed workflow can be particularly beneficial for small scale high-throughput pH adjustment experiments as required by R&D facilities in formulation chemistry and may incentivize data accumulation and management for pH adjustment processes. Moreover, we see a great potential of this technique in the age of personalized cosmetics and medicine as well as all other small batch formulation processes.

Acknowledgments

A. Pomberger is grateful for funding of his PhD studies by BASF SE and EPSRC Centre for Doctoral Training “The Automated Chemical Synthesis Enabled by Digital Molecular Technologies” (SynTech CDT, EP/S024220). This work is part co-funded by the EPSRC project “Combining Chemical Robotics and Statistical Methods to Discover Complex Functional Products" (EP/R009902). We thank Dr Connor Taylor, Kobi Felton, Daniel Wigh and Ferdinand Kossmann for helpful discussions and providing feedback on the manuscript. We thank K. U. Thum for support with the experimental work. This work is co-funded by National Research Foundation (Singapore) grant “Cambridge Centre in Carbon Reduction Technologies”, via CREATE programme, and by European Regional Development Fund, via the project “Innovation Centre in Digital Molecular Technologies”.

Conflict of interest

The authors declare no conflict of interest.

References

1. Michl, J.; Park, K. C.; Swietach, P., Evidence-based guidelines for controlling pH in mammalian live-cell culture systems. *Communications Biology* **2019,** *2* (1), 144.

2. Alwan, G. M., pH-Control Problems of Wastewater Treatment Plants. *Al-Khwarizmi Engineering Journal,* **2008,** *4* (2), 37-45.

3. Goel, R. K.; Flora, J. R. V.; Chen, J. P., Flow Equalization and Neutralization. In *Physicochemical Treatment Processes. Handbook of Environmental Engineering*, 2005; Vol. 3, pp 22-26.

4. Lukić, M.; Pantelić, I.; Savić, S. D., Towards Optimal pH of the Skin and Topical Formulations: From the Current State of the Art to Tailored Products. *Cosmetics* **2021,** *8* (3), 69.

5. Hawkins, S.; Dasgupta, B. R.; Ananthapadmanabhan, K. P., Role of pH in skin cleansing. *International Journal of Cosmetic Science* **2021,** *43* (4), 474-483.

6. Kalak, T.; Gąsior, K.; Wieczorek, D.; Cierpiszewski, R., Improvement of washing properties of liquid laundry detergents by modification with N-hexadecyl-N,N-dimethyl-3-ammonio-1-propanesulfonate sulfobetaine. *Textile Research Journal* **2020,** *91* (1-2), 115-129.

7. Tan, W. W.; Lu, F.; Loh, A. P.; Tan, K. C., Modeling and control of a pilot pH plant using genetic algorithm. *Engineering Applications of Artificial Intelligence* **2005,** *18* (4), 485-494.

8. Hasselbalch, K. A., Die Berechnung der Wasserstoffzahl des Blutes aus der freien und gebundenen Kohlensaeuure desselben, und die Sauerstoffbindung des Blutes als Funktion der Wasserstoffzahl. *Biochemische Zeit* **1916,** (78), 112-144.

9. Nguyen, M. K.; Kao, L.; Kurtz, I., Calculation of the equilibrium pH in a multiple-buffered aqueous solution based on partitioning of proton buffering: a new predictive formula. *American Journal of Physiology-Renal Physiology* **2009,** *296* (6), F1521-F1529.

10. Bennett, S., A brief history of automatic control. *IEEE Control Systems Magazine* **1996,** *16* (3), 17-25.

11. Zhu, Y.; Nishigori, S.; Shimura, N.; Nara, T.; Fujimori, E., Development of an Automatic pH Adjustment Instrument for the Preparation of Analytical Samples Prior to Solid Phase Extraction. *Analytical Sciences* **2020,** *36* (5), 621-626.

12. Imtiaz, U.; Jamuar, S. S.; Sahu, J. N.; Ganesan, P. B., Bioreactor profile control by a nonlinear auto regressive moving average neuro and two degree of freedom PID controllers. *Journal of Process Control* **2014,** *24* (11), 1761-1777.

13. Harcum, S. W.; Elliott, K. S.; Skelton, B. A.; Klaubert, S. R.; Dahodwala, H.; Lee, K. H., PID controls: the forgotten bioprocess parameters. *Discover Chemical Engineering* **2022,** *2* (1), 1-18.

14. Chotteau, V.; Hjalmarsson, H. In *Tuning of Dissolved Oxygen and pH PID Control Parameters in Large Scale Bioreactor by Lag Control*, Proceedings of the 21st Annual Meeting of the European Society for Animal Cell Technology (ESACT), , 2009; pp 327-330.

15. Hoshan, L.; Jiang, R.; Moroney, J.; Bui, A.; Zhang, X.; Hang, T.-C.; Xu, S., Effective bioreactor pH control using only sparging gases. *Biotechnology Progress* **2019,** *35* (1), 1-7.

16. Altınten, A., Generalized predictive control applied to a pH neutralization process. *Computers & Chemical Engineering* **2007,** *31* (10), 1199-1204.

17. Helmy, H.; Janah, D. A. M.; Nursyahid, A.; Mara, M. N.; Setyawan, T. A.; Nugroho, A. S. In *Nutrient Solution Acidity Control System on NFT-Based Hydroponic Plants Using*

Multiple Linear Regression Method, 2020 7th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), pp 272-276.

18. Alkamil, E. H. K.; Al-Dabooni, S.; Abbas, A. K.; Flori, R.; Wunsch, D. C., Learning From Experience: An Automatic pH Neutralization System Using Hybrid Fuzzy System and Neural Network. *Procedia Computer Science* **2018,** *140*, 206-215.

19. He, N.; Zhang, M.; Li, R., An Improved Approach for Robust MPC Tuning Based on Machine Learning. *Mathematical Problems in Engineering* **2021,** *2021*, 1-18.

20. Åkesson, B. M.; Toivonen, H. T.; Waller, J. B.; Nyström, R. H., Neural network approximation of a nonlinear model predictive controller applied to a pH neutralization process. *Computers & Chemical Engineering* **2005,** *29* (2), 323-335.

21. Pan, S. J.; Yang, Q., A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering* **2010,** *22*, 1345-1359.

22. Pesciullesi, G.; Schwaller, P.; Laino, T.; Reymond, J.-L., Transfer learning enables the molecular transformer to predict regio- and stereoselective reactions on carbohydrates. *Nature Communications* **2020,** *11* (1), 1-8.

23. Zhang, Y.; Wang, L.; Wang, X.; Zhang, C.; Ge, J.; Tang, J.; Su, A.; Duan, H., Data augmentation and transfer learning strategies for reaction prediction in low chemical data regimes. *Organic Chemistry Frontiers* **2021,** *8* (7), 1415-1423.

24. Amar, Y.; Schweidtmann, Artur M.; Deutsch, P.; Cao, L.; Lapkin, A., Machine learning and molecular descriptors enable rational solvent selection in asymmetric catalysis. *Chemical Science* **2019,** *10* (27), 6697-6706.

25. Zhang, C.; Amar, Y.; Cao, L.; Lapkin, A. A., Solvent Selection for Mitsunobu Reaction Driven by an Active Learning Surrogate Model. *Organic Process Research & Development* **2020,** *24* (12), 2864-2873.

26. Mohri, M.; Rostamizadeh, A.; Talwalkar, A., *Foundations of machine learning*. MIT press: 2012.

27. Jordan, M. I.; Mitchell, T. M., Machine learning: Trends, perspectives, and prospects. *Science* **2015,** *349* (6245), 255-260.

28. Carleo, G.; Cirac, I.; Cranmer, K.; Daudet, L.; Schuld, M.; Tishby, N.; Vogt-Maranto, L.; Zdeborová, L., Machine learning and the physical sciences. *Reviews of Modern Physics* **2019,** *91* (4), 1-39.

29. Eyke, N. S.; Green, W. H.; Jensen, K. F., Iterative experimental design based on active machine learning reduces the experimental burden associated with reaction screening. *Reaction Chemistry & Engineering* **2020,** *5* (10), 1963-1972.

30. Jorayev, P.; Russo, D.; Tibbetts, J. D.; Schweidtmann, A. M.; Deutsch, P. P.; Bull, S. D.; Lapkin, A. A., Multi-objective Bayesian optimisation of a two-step synthesis of p-cymene from crude sulphate turpentine. *Chemical Engineering Science* **2021**, 116938 1-10.

31. Shields, B. J.; Stevens, J.; Li, J.; Parasram, M.; Damani, F.; Alvarado, J. I. M.; Janey, J. M.; Adams, R. P.; Doyle, A. G., Bayesian reaction optimization as a tool for chemical synthesis. *Nature* **2021,** *590* (7844), 89-96.

32. Ho, T. K., Random decision forests. *Proceedings of 3rd International Conference on Document Analysis and Recognition* **1995,** *1*, 278-282.

33. Rasmussen, C. E.; Williams, C. K. I., Gaussian Processes for Machine Learning. *MIT Press* **2006**.

34. Schmidhuber, J., Deep learning in neural networks: An overview. *Neural Networks* **2015,** *61*, 85-117.

35. Cao, L.; Russo, D.; Felton, K.; Salley, D.; Sharma, A.; Keenan, G.; Mauer, W.; Gao, H.; Cronin, L.; Lapkin, A. A., Optimization of Formulations Using Robotic Experiments Driven by Machine Learning DoE. *Cell Reports Physical Science* **2021,** *2* (1), 100295 1-17.
36. Salley, D. S.; Keenan, G. A.; Long, D.-L.; Bell, N. L.; Cronin, L., A Modular Programmable Inorganic Cluster Discovery Robot for the Discovery and Synthesis of Polyoxometalates. *ACS Central Science* **2020,** *6* (9), 1587-1593.
37. Nicolas, J. FLab. https://pypi.org/project/flab/ (accessed 23.04.22).
38. Loh, W.-Y., Regression Trees With Unbiased Variable Selection and Interaction Detection. *Statistica Sinica* **2002,** *12*, 361-386.
39. Wigh, D. S.; Goodman, J. M.; Lapkin, A. A., A review of molecular representation in the age of machine learning. *WIREs Computational Molecular Science*, **2022**, e1603, 1-19.
40. Pomberger, A.; Pedrina McCarthy, A. A.; Khan, A.; Sung, S.; Taylor, C. J.; Gaunt, M. J.; Colwell, L.; Walz, D.; Lapkin, A. A., The effect of chemical representation on active machine learning towards closed-loop optimization. *Reaction Chemistry & Engineering* **2022,** *7*, 1368-1379.

Automated pH Adjustment Driven by Robotic Workflow and Active Machine Learning

Supporting Information

A. Pomberger[1,2], N. Jose[1,2,3,4], D. Walz[5], J. Meissner[5], C. Holze[5], Matthäus Kopczynski[5], P. Müller-Bischof[6] and A. A. Lapkin[1,2,3]

[1]Department of Chemical Engineering and Biotechnology, University of Cambridge, CB3 0AS Cambridge, United Kingdom
[2]Innovation Centre in Digital Molecular Technologies (iDMT), Yusuf Hamied Department of Chemistry, University of Cambridge, CB2 1EW Cambridge, United Kingdom
[3] Cambridge Centre for Advanced Research and Education in Singapore, CARES Ltd. 1 CREATE Way, CREATE Tower, 138602 Singapore
[4] Accelerated Materials Ltd, 71-75, Shelton Street, WC2H 9JK London, United Kingdom
[5] BASF SE, Carl-Bosch-Strasse 38, 67056 Ludwigshafen am Rhein, Germany
[6] Vienna University of Technology, Karlsplatz 13, 1040 Vienna, Austria

Table of Contents

Materials

Unless mentioned otherwise, all solvents and chemicals were purchased from commercial suppliers and were used as received. Compound names are based following the IUPAC nomenclature.

For initial manual experiments a Metrohm 716 DMS Titrino was used and was calibrated via three-point calibration using buffers of pH 4, 7 and 10. For the automated pH measurement studies the pH meter (VWR Model 662-1767) was also calibrated with the latter buffers.

Development of the Robotic Platform

The requirement to generate experimental training data as well as to have an automated experimental pH adjustment device led us develop a robotic platform, based on previously reported literature.[1, 2] As shown in Figure S1a, the setup consists of a robotic wheel, containing up to 24 sample vials (each up to 15 mL), 10 syringe pumps (TriContinent Model 7026-01), a robotic arm and consisting of two stepper motors to facilitate horizontal and vertical movement (Servo Tecnico 1005SNSN-001). The robotic arm was used to move the pH electrode (VWR Model 662-1767) into the sample vial as well as to the cleaning positions to avoid cross contamination across the samples, see Figure S1b. Moreover, the pH electrode has the dosing tubes for acid and base attached, to allow for direct dosing into the sample and immediate pH measurement. The cleaning position consists of a continuously flushed vial which is connected to a peristaltic pump, providing deionized water, see Figure S1c. The pumps were connected to the stock solutions (using PFA tubing) and to the dosing position on the wheel, see Figure S2d. Stepper motors and pH electrode were controlled using Arduino (Board model: Mega 2560) whereas the pumps were controlled directly via their serial port, relying on FLab for communication over all components. Since part of the experiments as well as troubleshooting were conducted remotely, we relied on a webcam to monitor the robot's status. Magnetic stirrers (fans modified with magnets) were placed below the cleaning position and the position where the pH measurement was conducted to facilitate sufficient mixing of the sample, particularly upon acid/base addition. For more information on the technical specifications see Cao et al. [1]

The platform can be operated without coding experience in screening mode where a spreadsheet sheet or a csv file with information about the components (pump volume of the stock solution) well as the information about the titration strategy (acid/base, steps, volumes per step) is submitted to the robotic platform and all experiments are executed without human intervention. Finally, a csv file containing the data of the experiments is received as an output and can be directly used for ML modelling purposes.

In order to use the platform for automated pH adjustment, we relied on FLab to facilitate the communication between the single components and the ML based optimization strategy, allowing for walk-away conditions once the target pH was set.

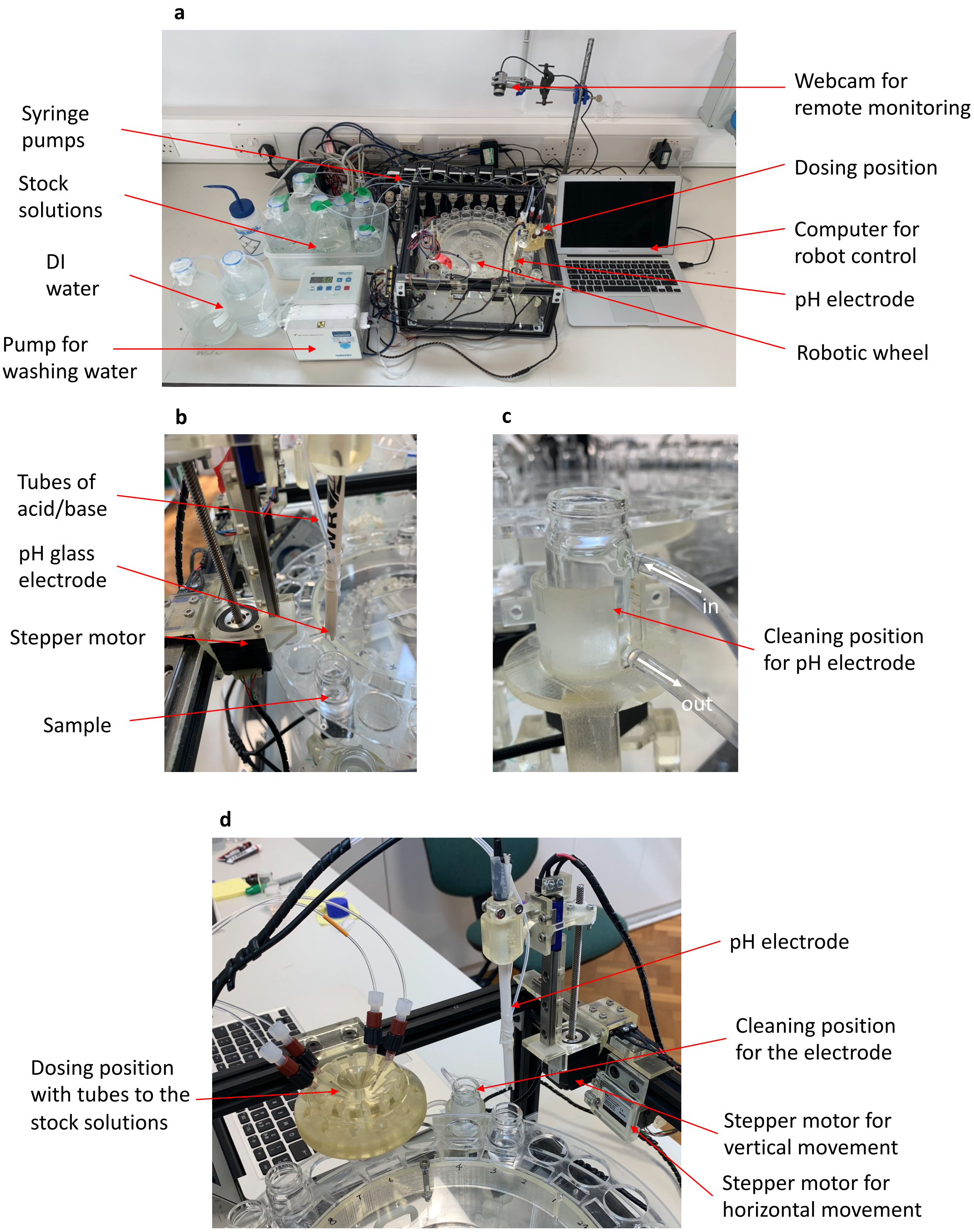

a
Syringe pumps
Stock solutions
DI water
Pump for washing water
Webcam for remote monitoring
Dosing position
Computer for robot control
pH electrode
Robotic wheel
b
Tubes of acid/base
pH glass electrode
Stepper motor
Sample
c
in
out
Cleaning position for pH electrode
d
pH electrode
Cleaning position for the electrode
Dosing position with tubes to the stock solutions
Stepper motor for vertical movement
Stepper motor for horizontal movement

Figure S1. Robotic platform used for conducting automatic pH adjustment. (a) Overall view of the system (B) Detailed view of the pH electrode (d) Detailed view of the washing position (d) Detailed view of the dosing position and pH electrode.

Machine Learning Models

This section details the identification of suitable hyperparameters of the surrogate models. As opposed to traditional hyperparameter tuning – where the objective is to find the parameters that deliver a low prediction error – we aimed to decrease the average number of iterations to reach the target pH within the active ML-driven closed-loop optimization. The presented values are the mean (Eqn. 1) and standard deviation of 10 single experiments, to allow for generalizability. Error bars are reported in error on the mean value, see Eqn. 2.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \qquad \text{Eqn. S1}$$

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{N-1}} \qquad \text{Eqn. S2}$$

where

σ: standard deviation

x_i: single observation

μ: mean value

N: number of observations

Preliminary Studies - Extrapolative predictions

We started with an externally generated pH dataset, containing acid and base titration information of 18 binary mixtures, giving a dataset of 1956 single datapoints (single pH measurements). Due to the density of datapoints, a simple random split of the data – where the data is split up randomly in training and test data – would not allow to understand whether the ML model is able to perform useful predictions for realistic applications, such as extremely low amount of data. Thus, we selectively chose the data which is present in the test and training partition and designed the preliminary experiments as extrapolative prediction tasks.

We trained ML models on the experimental titration data of the pure chemicals (e.g., ammonium and KH_2PO_4) and evaluated ML predictions for the binary mixture (experimental data was existent), thus performing an extrapolative prediction. The large feature set, using chemical information was used. Figure S2 illustrates true vs predicted titration curves – as visible the predictions are capable of predicting the trend, however, often clearly miss the ground truth.

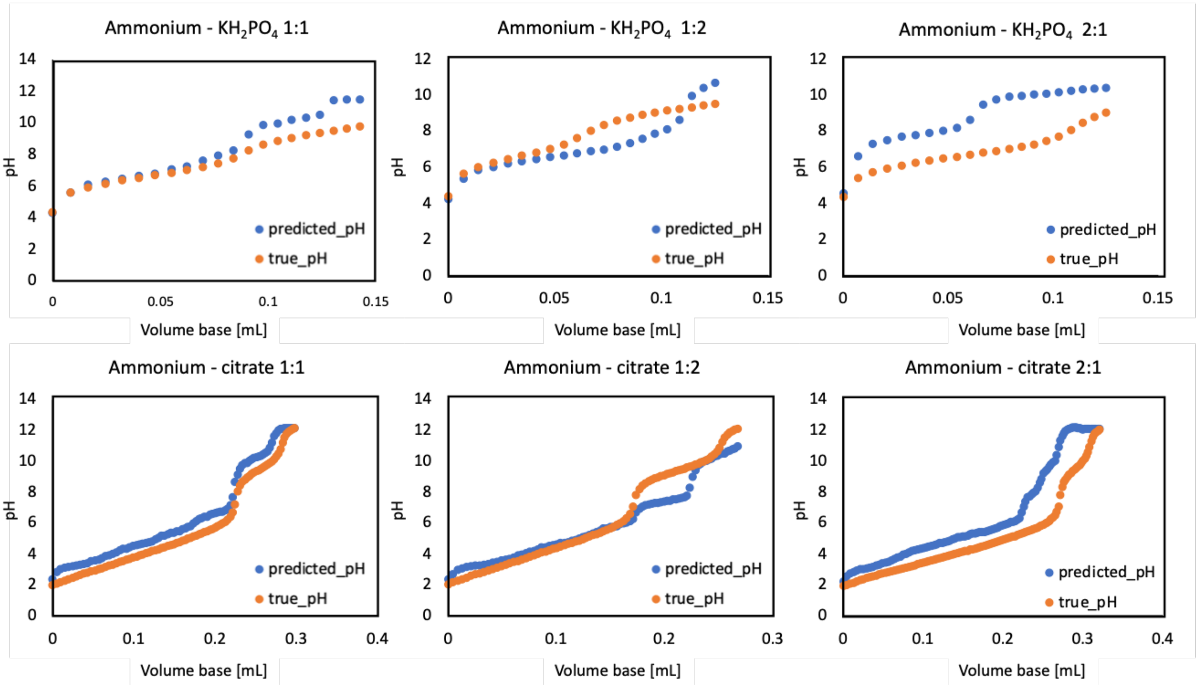


Figure S2. Comparison of predicted vs true pH curves of given systems using base titration data. Each datapoint represents a single measurement/prediction, the difference in datapoint density is due to the experimental workflow.

While we initially attempted to develop a purely predictive model for extrapolation across different systems, we understood the underlying challenges and limitations of accurately modelling the pH of multi-buffered mixtures within extrapolative predictions. Eventually, we changed our strategy from a purely predictive approach to an iterative strategy, involving repeated sampling and model training.

Surrogate Models and Hyperparameter Optimization

In traditional ML hyperparameter optimization the data is split up into train and test partitions via random split or cross-validation methods to further conduct hyperparameter tuning. In this study we aim to optimize the hyperparameters of the models specifically for the target task – active ML driven closed-loop optimization. This approach was applied since the hyperparameters need to be tuned according to the same low-data regime which will be required for subsequent applications within pH adjustment – using larger datasets and conducting train/test splits would not be relevant as this strategy differs from the target task. Objective of the hyperparameter tuning was to find configurations that allow the target task – pH adjustment – find the objective in the minimum number of iterations. This section details the process for the used surrogate models and provides additional information on the implementation.

Artificial Neural Net

The artificial neural network model was implemented using Tensorflow Keras 2.3.0. The finally used fully connected feed-forward network consists of three hidden layers of 40 nodes each and ELU activation function. The final layer consisted of one single node. The weights were initialized with the default schemes (Glorot uniform and zeros, respectively). Training was done with RMSProp using default parameters over 1000 epochs with a minibatch size of 32 and a learning rate of 0.015.

The results of the hyperparameter optimization can be found below.

Learning Rate

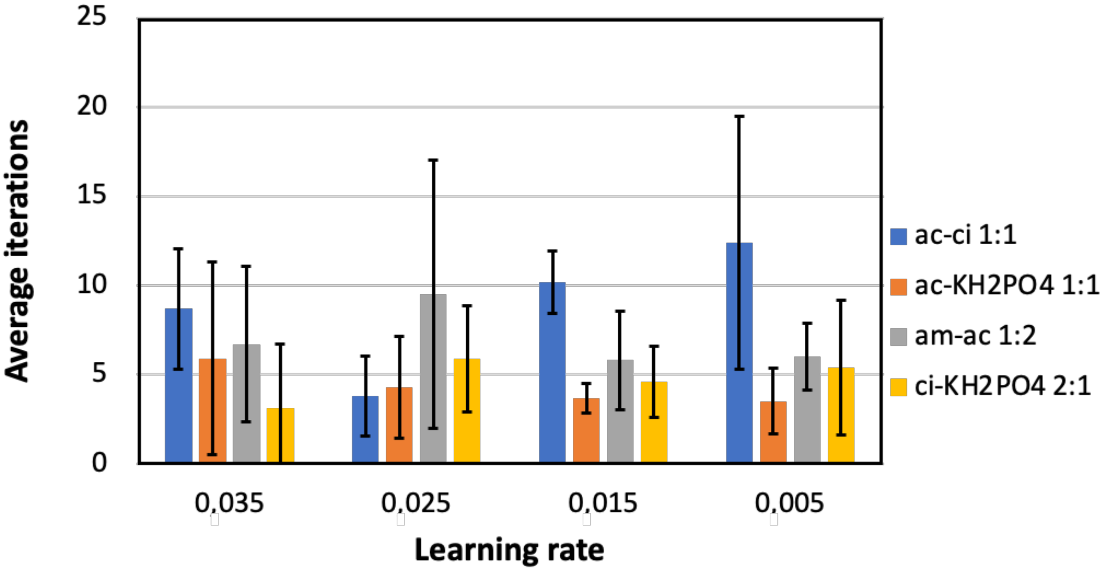


Figure S3. Comparison of different learning rates for the ANN. 10 single iterations, error bars are reported according to Eqn. 2.

Table S1. Summary of the Result of learning rate experiments for ANN.

Learning rate	**ac-ci (1:1)**	**ac-KH_2PO_4 (1:1)**	**am-ac (1:2)**	**ci-KH_2PO_4 (2:1)**	**average**
0.035	8.7	5.9	6.7	3.1	6.10
0.025	3.8	4.3	9.5	5.9	5.87
0.015	10.2	3.7	5.8	4.6	6.07
0.005	12.4	3.5	6.0	5.4	6.83

The average value of each learning rate was calculated and compared to find the rate with the least iterations. Although a learning rate of 0.025 had the least iterations, we decided to continue with a learning rate of 0.015 due to the smaller standard deviation, see Figure S3.

Epochs

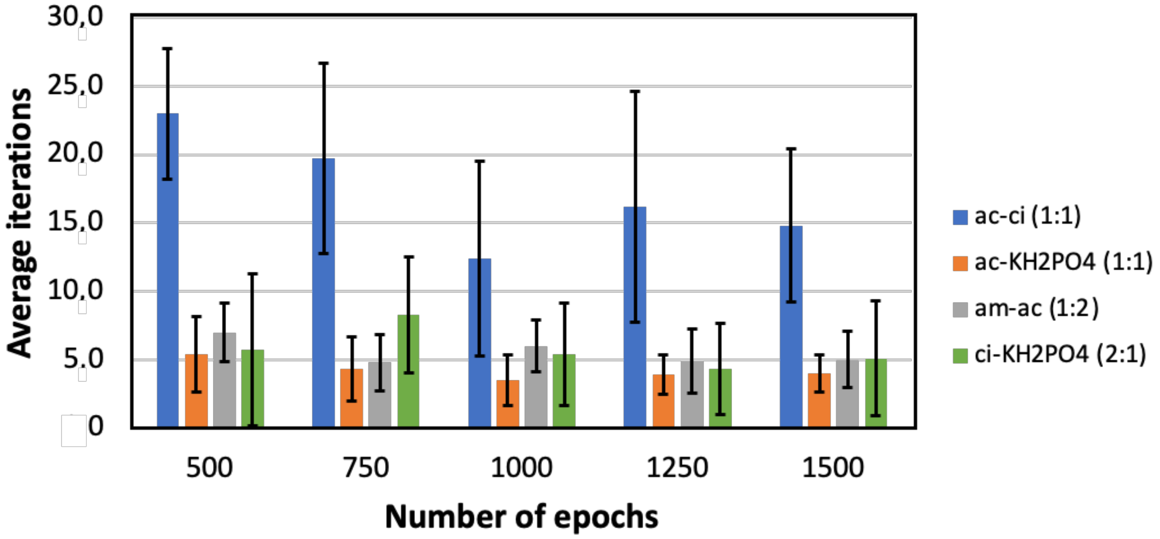


Figure S4. Comparison of different epoch values for the ANN. 10 single iterations, error bars are reported according to Eqn. 2.

Table S2. Summary of result of epoch experiments for ANN.

Epochs	**ac-ci (1:1)**	**ac-KH_2PO_4 (1:1)**	**am-ac (1:2)**	**ci-KH_2PO_4 (2:1)**	**mean**
500	23.0	5.4	7.0	5.7	10.28
750	19.7	4.3	4.8	8.3	9.28
1000	12.4	3.5	6.0	5.4	6.83
1250	16.2	3.9	4.9	4.3	7.33
1500	14.8	4.0	5.0	5.1	7.23

The average for each learning rate was calculated and compared to find the one with the least iterations. A value of 1000 epochs gave the best result across the three chosen datasets.

Activation Function

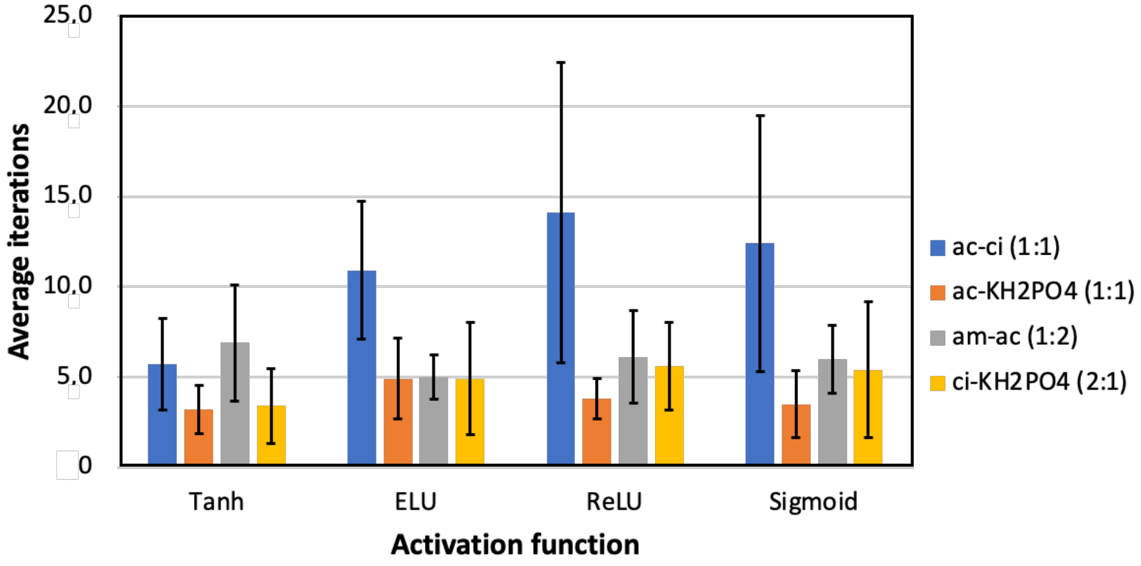


Figure S5. Comparison of different activation functions for the ANN. 10 single iterations, error bars are reported according to Eqn. 2.

Table S3. Summary of the results of activation function experiment for the ANN.

AF	**ac-ci (1:1)**	**ac-KH_2PO_4 (1:1)**	**am-ac (1:2)**	**ci-KH_2PO_4 (2:1)**	**mean**
Tanh	5.7	3.2	6.9	3.4	4.80
ELU	10.9	4.9	5.0	4.9	6.43
ReLU	14.1	3.8	6.1	5.6	7.40
Sigmoid	12.4	3.5	6.0	5.4	6.83

The activation function Tanh gave the best results. However, it performed very low during the benchmark in combination with all optimized parameters. Therefore, we decided to use the second-best activation function, ELU which delivered a better outcome.

Number of Neurons

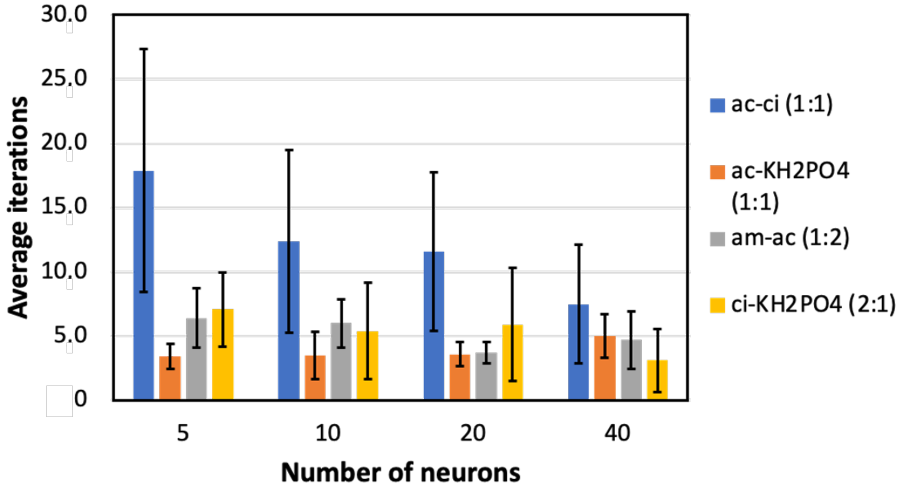


Figure S6. Comparison of different neuron numbers per layer for ANN. 10 single iterations, error bars are reported according to Eqn. 2.

Table S4. Summary of the results of the neuron number experiment for ANN.

Neurons	**ac-ci (1:1)**	**ac-KH_2PO_4 (1:1)**	**am-ac (1:2)**	**ci-KH_2PO_4 (2:1)**	**mean**
5	17.9	3.4	6.4	7.1	8.70
10	12.4	3.5	6.0	5.4	6.83
20	11.6	3.6	3.7	5.9	6.20
40	7.5	5.0	4.7	3.1	5.08

Due to computational expenses, we stopped at 40 neurons per layer – this value also delivered a suitable performance.

Number of Layers

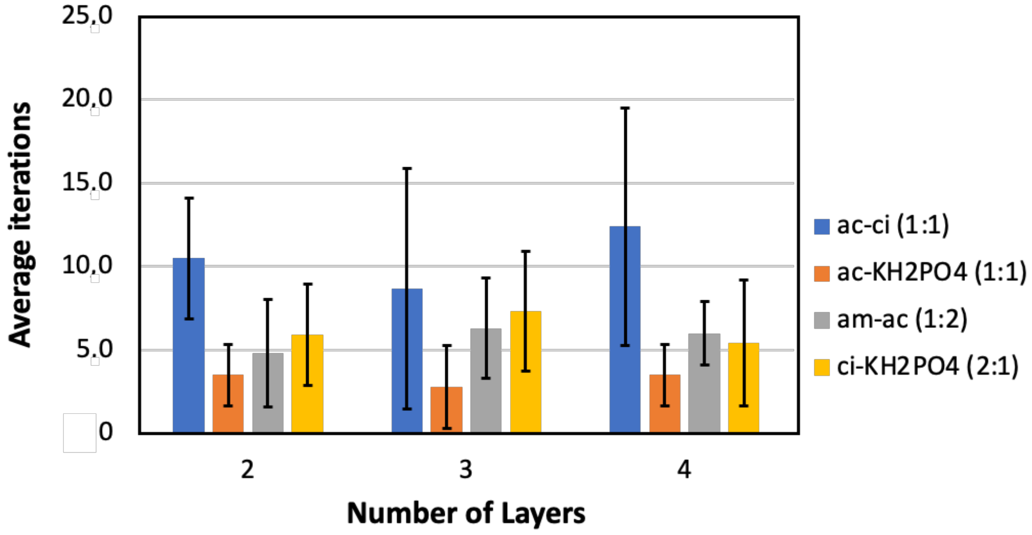


Figure S7. Comparison of different number of hidden layers for ANN. 10 single iterations, error bars are reported according to Eqn. 2.

Table S5. Summary of the results of hidden layer experiments for ANN.

Hidden layers	**ac-ci (1:1)**	**ac-KH_2PO_4 (1:1)**	**am-ac (1:2)**	**ci-KH_2PO_4 (2:1)**	**mean**
2	10.5	3.5	4.8	5.9	6.18
3	8.7	2.8	6.3	7.3	6.28
4	12.4	3.5	6.0	5.4	6.83

According to the table above, it seems that 2 hidden layers gave the best result. However, in combination with the optimized value for the number of neurons per layer (40 neurons) the accuracy decreased significantly. For this reason, we decided to increase to 3 hidden layer, which better results.

Gaussian Process

We implemented the Gaussian process surrogate model using the package scikitlearn, version 0.23.0. In terms of the kernel, the empirically tested kernels can be found below, see Table S7. For the main analysis we used RBF kernel with a lengthscale limited to the range 0.1 – 1000. No white kernel was added to account for measurement noise as we did not observe a performance boost.

Kernel Function

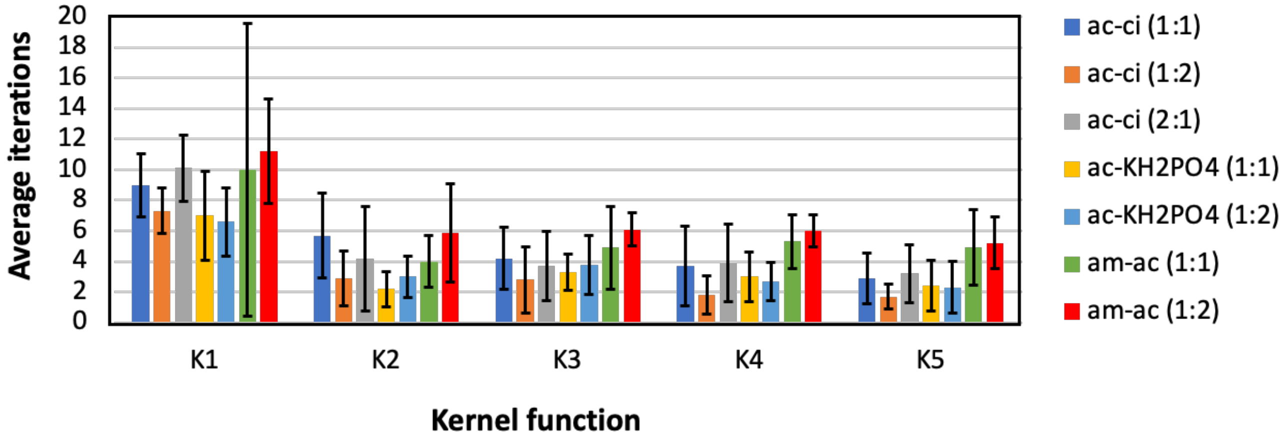


Figure S8. Comparison of different kernels of the GP using seven training datasets. 10 single iterations, error bars are reported according to Eqn. 2.

Table S6. Results of kernel experiments.

	ac-ci (1:1)	ac-ci (1:2)	ac-ci (2:1)	ac-KH_2PO_4 (1:1)	ac-KH_2PO_4 (1:2)	am-ac (1:1)	am-ac (1:2)	mean
K1	9.0	7.3	10.1	7.0	6.6	10.0	11.2	8.74
K2	5.7	2.9	4.2	2.2	3.0	4.0	5.9	3.99
K3	4.2	2.8	3.7	3.3	3.8	4.9	6.1	4.11
K4	3.7	1.8	3.9	3.0	2.7	5.3	6.0	3.77
K5	2.9	1.7	3.2	2.4	2.3	4.9	5.2	3.23

Table S7. Details of the used kernel functions.

K1	C(1.0) * (RBF(1,1e-10) + DotProduct())
K2	C(0.1, (1e-5, 1e2)) * RBF(100, (1e-3, 1e5))+ RBF(12, (1e-3, 1e5)) +RBF(1, (1e-3, 1e3))
K3	C(0.1, (1e-5, 1e2)) * RBF(1, (1e-3, 1e3))
K4	C(1) * RBF(1, (1e-3, 1e3))
K5	C(1) * RBF(1, (1e-1, 1e3))

Random Forest

We implemented the random forest surrogate model using the package scikit learn, version 0.23.0 – this version was used for all subsequent modelling. Based on preliminary insights the suitable number of estimators was found to be 400. All other parameters were kept as default.

Number of Estimators

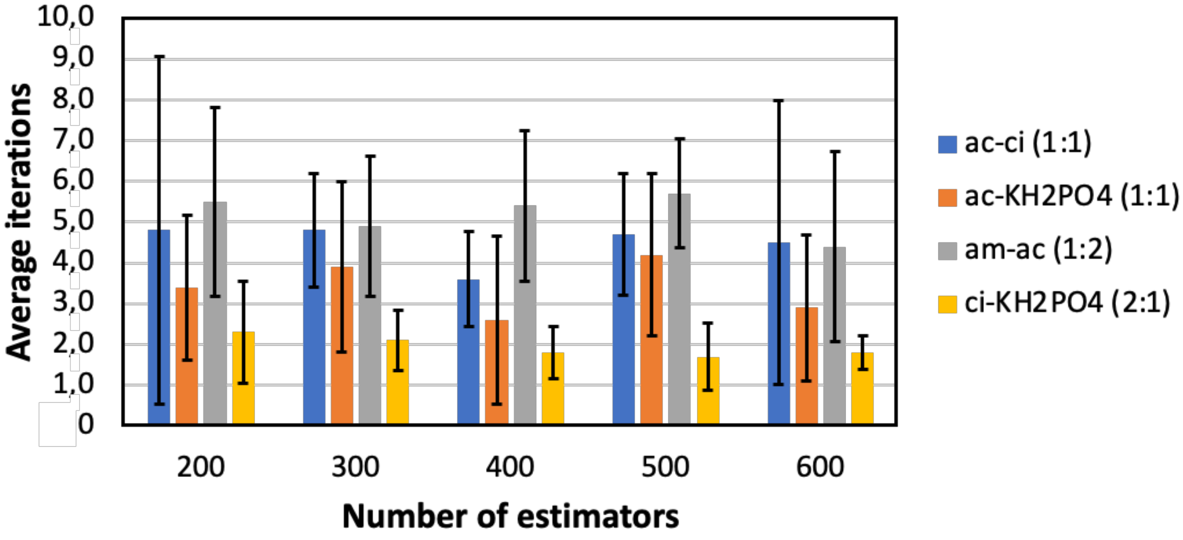


Figure S9. Comparison of different numbers of estimators for random forest model. 10 single iterations, error bars are reported according to Eqn. 2.

Estimators	**ac-ci (1:1)**	**ac-KH2PO4 (1:1)**	**am-ac (1:1)**	**Ci-KH2PO4 (1:2)**	**mean**
200	4.8	3.4	5.5	2.3	4
300	4.8	3.9	4.9	2.1	3.9
400	3.6	2.6	5.4	1.8	3.4
500	4.7	4.2	5.7	1.7	4.1
600	4.5	2.9	4.4	1.8	3.4

References

1. Cao, L.; Russo, D.; Felton, K.; Salley, D.; Sharma, A.; Keenan, G.; Mauer, W.; Gao, H.; Cronin, L.; Lapkin, A. A., Optimization of Formulations Using Robotic Experiments Driven by Machine Learning DoE. *Cell Reports Physical Science* **2021,** *2* (1), 100295 1-17.

2. Salley, D. S.; Keenan, G. A.; Long, D.-L.; Bell, N. L.; Cronin, L., A Modular Programmable Inorganic Cluster Discovery Robot for the Discovery and Synthesis of Polyoxometalates. *ACS Central Science* **2020,** *6* (9), 1587-1593.