Bayesian Optimization and in Organic Synthesis: Maximizing Reaction Yields

Arsenii Stolbov, Hannah Shiang, Zivai Sinemani

Abstract

Optimizing reaction conditions to maximize yield in organic synthesis chemistry is challenging, often requiring extensive experimentation that takes a lot of time and resources. An advanced optimization method that can be used to reduce the challenges is Bayesian Optimization (BO). This study investigates the effectiveness of BO, using Gaussian Process (GP) models, by comparing it with a simpler Multiple Linear Regression (MLR) model by optimizing the yield of a palladium-catalyzed direct arylation reaction. BO, particularly with the Expected Improvement (EI) acquisition function, demonstrated superior predictive accuracy and significantly reduced experimental workload in contrast with the MLR method. We found that BO has the potential to streamline reaction yield optimization while offering a scalable and efficient framework for complex chemical systems. This study not only emphasizes the practical potential of BO in computational chemistry but also lays the groundwork for future research into its broader applications.

Introduction

The field of organic synthesis often involves the optimization of reaction conditions, a process that traditionally requires a large number of experiments. Optimizing reaction conditions in organic chemistry involves the evaluation of multiple parameters such as substrate, catalyst, reactant, solvent, concentration, temperature, and reactor type. The challenge is to navigate this multidimensional parameter space to determine the conditions that maximize reaction efficiency and product yield. Traditionally, this process involves extensive experimentation based on literature precedents and mechanistic understanding. However, time and material constraints often limit chemists from exploring a fraction of the possible reaction configurations during optimization [1].

The problem of optimization is not unique to chemistry. In machine learning, algorithms analyze patterns in data to make predictions or decisions, often requiring fine-tuning of hyperparameters to achieve optimal performance. This process, like response optimization, involves a balance between exploring unknown possibilities and exploiting known successful configurations [2]. Bayesian Optimization (BO), a global optimization algorithm, has become a powerful machine learning tool for optimizing computationally expensive functions. Its adaptive nature dynamically learns from past evaluations to refine the search process, which significantly reduces the number of evaluations required compared to traditional methods.

BO is particularly well suited for problems with a large number of steps and complex evaluations, such as optimizing reaction conditions in organic synthesis. It builds a probabilistic model to predict the behaviour of the target function, guiding the selection of conditions that maximize the probability of success. By balancing research and exploitation, BO provides efficient and systematic reaction optimization with fewer experiments. In addition, it allows parallel evaluation of multiple experiments, making it a useful tool for

accelerating chemical research [3]. In recent years, BO has been widely considered for solving complex optimization problems in chemistry [3].

BO is a response surface-based iterative algorithm designed to optimize expensive (with many steps) objective functions. Using a probabilistic surrogate model, typically a Gaussian Process (GP), BO predicts the behaviour of the objective function in parameter space and identifies the most promising points for the next evaluation. GP is the most commonly used statistical model in BO [4]. It models each input $x \in X$ as a Gaussian random variable, characterized by a mean function $\mu(x)$ and a variance function $\sigma^2(x)$. To ensure that predictions are smooth and sensible, GPs rely on correlations between neighbouring points in the input space. These correlations are modelled using a kernel function, which measures the similarity between input points [4]. One of the simplest and most widely used kernel functions is the squared exponential kernel:

Equation 1[4]: $k_{SE}(x_i, x_j) = exp(-\ ||x_i - x_j||/2l^2)$

In Eq.(1): $k_{SE}(x_i,\ x_j)$: The kernel function value that quantifies the similarity between the two input points $x_i,\ x_j$.

A higher value indicates that the points are more similar, while a lower value indicates they are less similar. l is the length scale, a hyperparameter controlling how far correlations extend in the input space. $x_i,\ x_j$: The input data points being compared. $||x_i - x_j||$: The Euclidean distance between the two input points $x_i,\ x_j$. It measures how far apart the two points are in the input space to define correlations between data points. This enables GPs to provide both a predicted value and an uncertainty estimate for any input. The predictive mean and variance are computed as:

Equation 2[4]: $\mu_t(x) = k*(K_t + \sigma^2_{noise} I)^{-1} y$

Equation 3[4]: $\sigma^2_t(x) = k_{**} - k_*(K_t + \sigma^2_{noise} I)^{-1} k^T_*$

Where $K_{t,(i,j)} = k(x_i,\ x_j)$ is the kernel matrix,
$k_{**} = k_t(x,\ x),\ and\ k_* = [k(x_1,\ x),\ k(x_2,\ x),\ \ldots,\ k(x_t,\ x)]$, where I is the identity matrix with the same dimensions as Kt, and σ^2_{noise} is the output noise standard deviation.

GPs are particularly effective in BO due to their ability to model uncertainty, allowing a balance between exploring new domains and using known high-performance domains. However, they are computationally intensive for large datasets, scaling as an $O(N^3)$, and are heavily dependent on kernel choice, which can limit performance if it does not fit the data[4]. These limitations emphasize the need to explore alternative surrogate models for broader scalability and efficiency. Despite these challenges, GPs remain an integral part of BO by providing both an average prediction and an estimate of uncertainty allowing BO to effectively choose experiments. This dual capability of prediction and uncertainty modelling allows BO to balance exploration (testing unknown areas) and exploitation (refining known high-performance areas). Once the GP model has been generated, it is used to create acquisition functions [5]. The main feature of BO is the acquisition functions that determine the strategy for selecting the next experiment. Two commonly used acquisition functions are Expected improvement (EI) and Thompson sampling (TS).

Expected Improvement (EI). This method prioritizes points in the parameter space that are most likely to improve upon the current best-known result. EI calculates the expected gain from evaluating a specific point, considering both the predicted mean and uncertainty provided by the GP. By focusing on regions with high potential for improvement, EI efficiently narrows the search space while maintaining a balance between exploration and exploitation [6].

Thompson Sampling (TS). This method takes a probabilistic approach by sampling potential functions from the GP and selecting points based on their likelihood of being optimal. TS naturally incorporates a balance between leveraging known data (exploitation) and exploring less certain regions of the parameter space (exploration). Unlike EI, which directly computes expected improvement, TS uses a randomized strategy, making it particularly useful in scenarios with highly uncertain parameter spaces or noisy data [7].

Both EI and TS use the ability of the GP to model complex, nonlinear relationships between response conditions and outcomes. The GP works in the backend of BO to provide a flexible framework for evaluating the behaviour of the objective function. It is particularly well suited for chemical optimization, where the relationship between parameters and outcomes is often complex and where the sample size is often small.

We want to optimize the yield of a desired product depending on a combinatorial set of hundreds of possible reaction conditions using BO. To do so, we first have to find the best acquisition function for optimizing reaction yields. Consulting literature on BO usage in the organic chemistry community led us to use the Experiment-Driven Bayesian Optimization (EDBO) package[3]. Using the EDBO package[8], we test the efficiency of EI and TS in combination with different initialization strategies (k-means, k-medoids and random). This is to identify the acquisition function and initialization method that most efficiently optimizes the reaction yield while minimizing the number of experiments required.

In addition to evaluating BO, we compare its performance to a simpler optimization method such as Multiple Linear Regression (MLR). MLR models the relationship between a dependent variable (reaction yield) and multiple independent variables (e.g., temperature, solvent type, concentration). To account for categorical variables, such as solvent and catalyst types, we use dummy variables. Dummy variables are binary (0 or 1) variables created to represent categorical data in regression models [9]. For instance, if a dataset contains three types of solvents, two dummy variables such as (x1, x2) are used to encode each solvent type. (0,0) would represent the presence of the first solvent, (1, x2) would represent the presence of the second solvent, and (x1, 1) would represent the presence of the third solvent. This ensures that the model can account for their influence without assigning an arbitrary numerical value. Similarly, for variables like temperature and solvent volume, dummy variables are used to restrict values to those present in the dataset (temperature = 90, 105, or 120°C; solvent volume = 98, 150, or 263 µL), reflecting realistic experimental conditions [9–11].

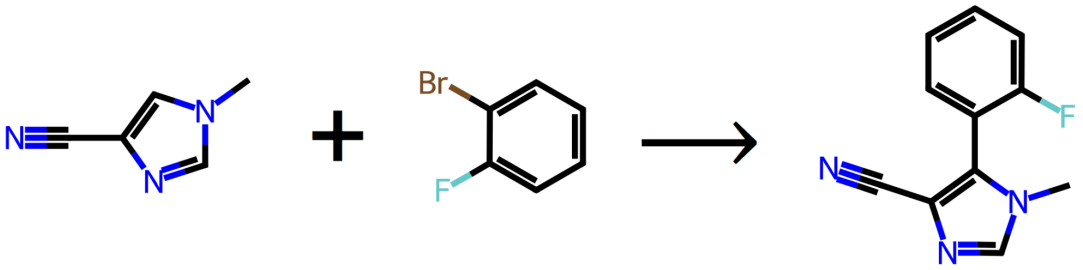
During this report, we aim to address the research question: Does BO significantly outperform simpler methods, such as MLR, for optimizing reaction yield? By comparing BO with MLR, we want to evaluate whether the advanced probabilistic framework of BO justifies its complexity in the context of reaction optimization. This systematic comparison provides insights into the strengths and limitations of each approach, offering a deeper understanding of how modern computational tools can enhance traditional workflows in synthetic chemistry.

Methodology

Part 1: Data Preparation

To use EDBO we need to find a data set that would fit the requirements of EDBO and be of interest to our study. The different reactions in the chosen dataset need to result in the same product and need to have percentage yield as a field.

Scheme 1: Imidazole Arylation Reaction[12].



We chose the palladium-catalyzed direct arylation reaction between imidazole and fluorobromoarene (Scheme 1). This reaction activates the C-H bond in imidazole, eliminating the need for pre-functionalized reagents. The result is a more efficient and environmentally friendly process with wide applications in the synthesis of biologically active compounds and pharmaceuticals [13]. The data set also has enough data points for us to test and all the necessary information needed for our experiment. Data for the palladium-catalyzed direct arylation reaction were obtained from the Open Reaction Database (ORD), a publicly available repository that compiles reaction datasets from the literature[12]. The chosen dataset includes a wide range of reaction conditions such as different reactants, solvents, catalysts, temperatures, and reaction yields. There are 256 reactions in total in the raw dataset. Preparing the dataset for compatibility with the EDBO package requires considerable preprocessing and adjustment due to inconsistencies in the format of the raw data and missing information.

The raw dataset is in a pb.gz file format which we converted to a JSON file which contains numerous missing entries and unnecessary data, which posed challenges for direct integration into the EDBO workflow. To combat these challenges, irrelevant and empty data columns were not included when converting the files to a CSV file. The columns in the final CSV file are Reagent SMILES, Solvent SMILES, Catalyst SMILES, reaction temperature, reagent amount, solvent volume, and catalyst amount. SMILES stands for Simplified Molecular Input Line Entry System and is a string format of molecular structures. However, the reagent and catalyst amounts are the same across all the reactions, thus, they were not considered.

In the end, we focused on five data columns that include a variety of values. The solvents include CC(N(C)C)=O, CCCC#N, CCCCOC(C)=O, and CC1=CC=C(C)C=C1. The catalysts include two palladium-based catalysts with different structural configurations which are P(C1=CC=CC=C1)(C2CCCCC2)C3=CC=CC=C3 and C1(P(C2CCCCC2)C3CCCCC3)=CC=CC=C1C4=CC=CC=C4. The reagent include O=C([O-])C.[K+], O=C([O-])C(C)(C)C.[K+], O=C([O-])C.[Cs+], and

O=C([O-])C(C)(C)C.[Cs+]. Solvent volumes are provided in microliters and include 98.039215, 150, and 263.1579. Temperatures present in the dataset are 120°C, 105°C, and 90°C.

There are several missing reactions that are needed to complete all possible combinations of the variables. We had to estimate the yields of these missing data points to fill in the gap. The data points missing are the ones with a temperature of 90 and a solvent volume of 150. We estimated the yields using the following process.

Step 1: Find and Scale the Differences

$For\ x\ =\ 105,\ 120;$

$D1\ =\ Temperature\ x's\ yield\ at\ volume\ 263\ -\ Temperature\ x's\ yield\ at\ volume\ 150;$
$D2\ =\ Temperature\ x's\ yield\ at\ volume\ 150\ -\ Temperature\ x's\ yield\ at\ volume\ 98;$

$SD1\ =\ D1/x's\ yield\ at\ volume\ 263;$
$SD2\ =\ D2/x's\ yield\ at\ volume\ 150;$

D1 = Difference 1;
D2 = Difference 2;
SD1 = Scaled Difference 1;
SD2 = Scaled Difference 2

If the yield is 0 for volume 263 or volume 98, use x's yield at volume 150. If D1 or D2 is 0 then SD1 or SD2 is equal to 0.

Step 2: Choose the Smallest Difference

$FSD1\ =\ Minimum\ between\ SD1_{105}\ and\ SD1_{120};$
$FSD2\ =\ Minimum\ between\ SD2_{105}\ and\ SD2_{120};$

FSD1 = Final Scaled Difference 1
FSD2 = Final Scaled Difference 2

Step 3: Find the Range Where the Missing Yield Percentage Could Exist

$Min\ =\ Yield\ at\ volume\ 98\ at\ temperature\ 90\ -\ [Yield\ at\ volume\ 98\ at\ temperature\ 90\ \times\ FSD2];$
$Max\ =\ Yield\ at\ volume\ 263\ at\ temperature\ 90\ +\ [Yield\ at\ volume\ 263\ at\ temperature\ 90\ \times\ FSD1]$

Min = The minimum value that the missing yield percentage can be
Max= The maximum value that the missing yield percentage can be

Step 4: Calculate the Missing Yield

Took yield for when the temperature is equal to 90 volume is equal to 150 as the mean of Min and Max.

$Missing\ Yield\ =\ (Min\ +\ Max)/2$

This process was done using Python on a ipynb file for all 32 missing yields. The 32 rows, including all the necessary columns in the original dataset, were then exported to a CSV file. One thing to keep in mind is that these are all estimates so there are some uncertainties with the final estimated values.

Part 2: Finding the Best Way to Run Bayesian Optimization (BO)

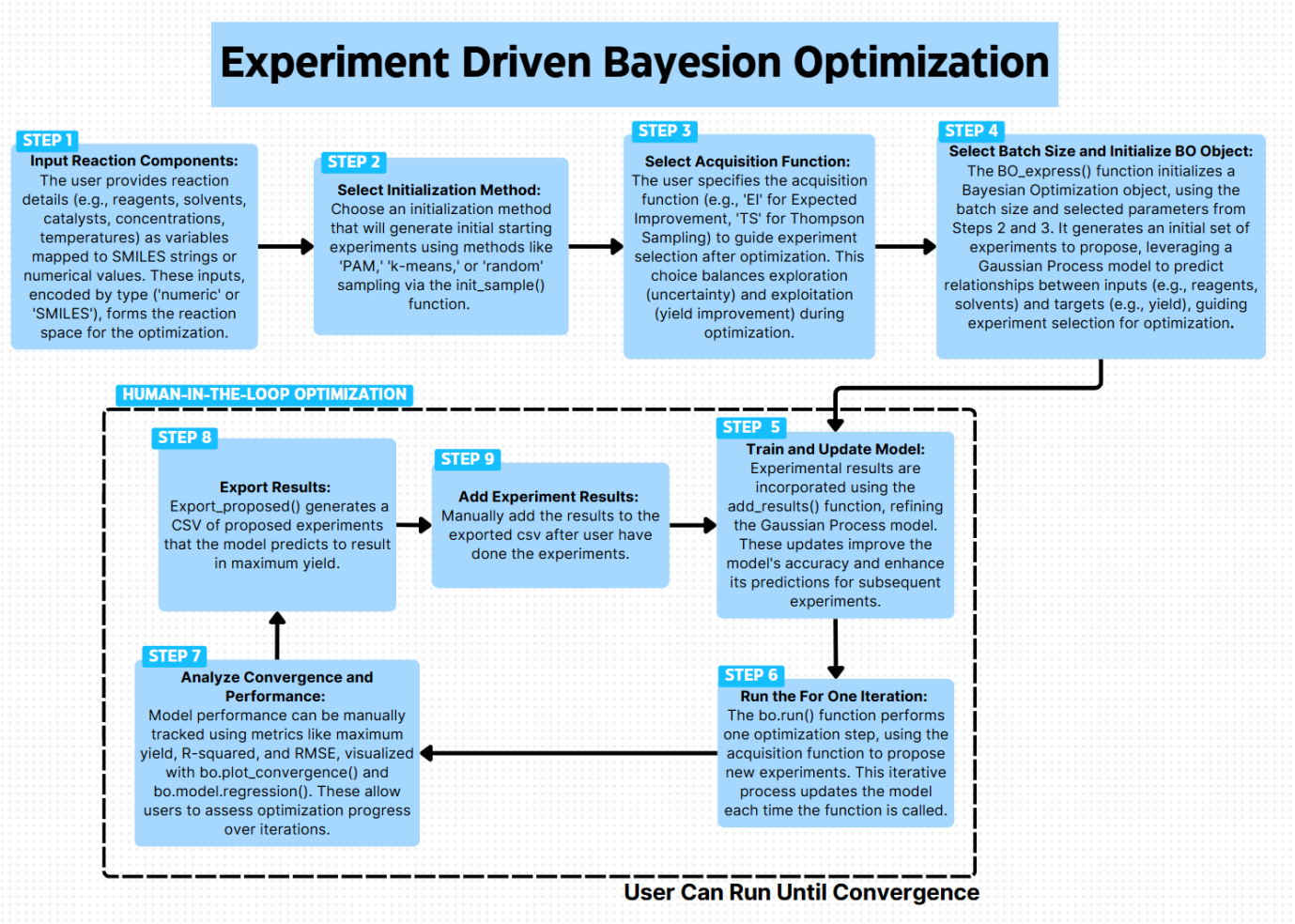


Figure 2: Flow Chart of EDBO Trial Process. Shows the step-by-step process of how EDBO works in running a BO trial.

The optimization process using EDBO begins with an initialization run to determine the starting experiments, utilizing one of three methods: k-means, k-medoids, or random selection.

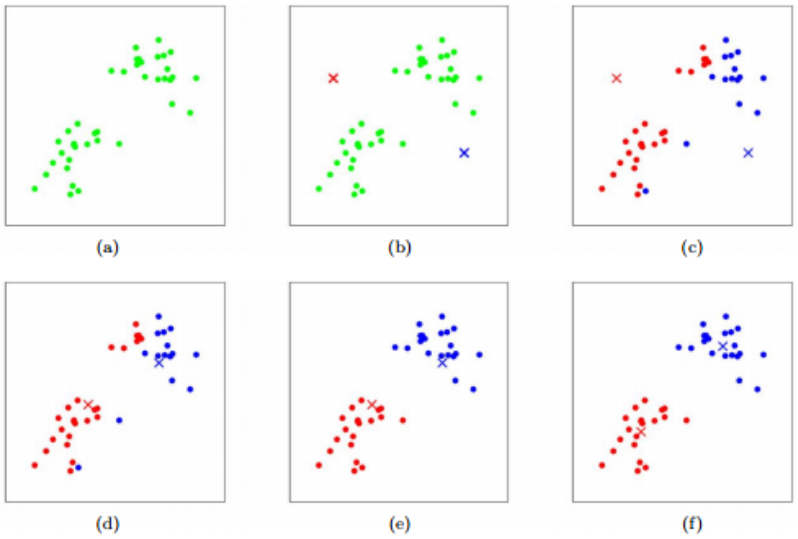


Figure 3: K-means Algorithm Representation [14]. The k-means algorithm is visually represented. (a) The initial scatter plot of the data point with no manipulation. (b) The selection of initial random centroids, denoted by "x", serves as the starting cluster centers. (c) The formation of initial clusters based on the centroids. (d), (e), and (f) The iterative process of updating centroids and clusters until the clusters stabilize, and the centroids converge to the center of each distinct cluster.

The k-means method, an unsupervised machine learning algorithm, clusters data into distinct groups by identifying optimal starting points. The algorithm works by initially selecting random center points, called centroids, which serve as the starting cluster centers. Data points are then assigned to the nearest centroid, forming clusters based on proximity. The centroids are iteratively updated by recalculating the mean position of the points within each cluster until the clusters stabilize and the centroids converge to their final positions. In the context of EDBO, the k-means initialization method identifies and selects initial experiments closest to the centroids, ensuring that each selected experiment represents the average value of a distinct cluster[14,15].

The k-medoids method operates similarly to k-means but differs by using actual data points as cluster centers, called medoids, rather than computed centroids. The process begins with the random selection of existing data points as medoids. Data points are then assigned to the nearest medoid to form clusters. The medoids are iteratively updated by minimizing the total distance between points and their medoid until the clusters stabilize. In the EDBO implementation, the k-medoids method returns the final medoids as the starting experiments, ensuring that each selected experiment corresponds to a real point from the dataset. This approach guarantees practical and representative initial conditions for the optimization[15,16].

For comparison, a random initialization method was also employed, where starting experiments are selected randomly without any clustering or analysis. This method provides a point of comparison for evaluating the more systematic approaches, such as k-means and k-medoids[1].

Following the initialization run, the optimization process focuses on selecting subsequent experiments using one of two acquisition functions: Expected Improvement (EI) or Thompson Sampling (TS). EI identifies points in the parameter space that are most likely

to yield improvements over the current best-known result, effectively balancing exploration and exploitation by incorporating both the predicted mean and uncertainty provided by the GP. In contrast, TS employs a probabilistic strategy, sampling potential functions from the GP and selecting points based on their likelihood of being optimal, making it particularly effective in uncertain or noisy parameter spaces.

The optimization workflow ran for four rounds of BO, with a batch of seven experiments per round. Reaction yield was set as the objective function, and each round leveraged the chosen acquisition functions to systematically propose experiments. This iterative approach facilitated an efficient and adaptive search for optimal reaction conditions while minimizing the total number of experiments required. The overall EDBO process was repeated for six trials using all the different combinations of the initialization methods and the aquisition functions.

The performance of each trial was evaluated using two key criteria: convergence and regression metrics. Convergence is assessed by analyzing the progression of the maximum observed yield versus optimization rounds, providing insight into the rate at which the optimization process approached the optimal solution. To quantify the accuracy of the predictive models, regression metrics including Root Mean Square Error (RMSE) and the coefficient of determination (R^2) were employed[17]. RMSE measures the average magnitude of prediction errors by calculating the square root of the mean of the squared differences between predicted and actual values.

Equation 4: $$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(P_i - O_i)^2}$$

In the Eq. (4):

P_i: represents the predicted value for the i^{th} observation,

O_i: denotes the observed value for the i^{th} observation,

n: is the total number of observations.

This metric provides a direct indication of the model's predictive accuracy, where a lower RMSE value indicates that the model predictions are closer to the actual values, indicating higher predictive accuracy (model fits the data).

The coefficient of determination (R^2) is a statistical measure that evaluates the proportion of variance in the dependent variable explained by the independent variables in a model. R^2 values range from 0 to 1, where an R^2 of 0 indicates that the model fails to explain any of the observed variance, and an R^2 of 1 signifies that the model perfectly captures all the variability in the dependent variable. Mathematically, R^2 is expressed as:

Equation 5: $$R^2 = 1 - (SS_{res} / SS_{tot})$$

In the Eq. (5):

SS_{res}: is the residual sum of squares, representing the variance unexplained by the model,

SS_{tot}: is the total sum of squares, representing the total variance in the observed data.

A higher R^2 value indicates that the model provides a better fit to the data, demonstrating its effectiveness in identifying and capturing the underlying patterns within the dataset. This makes R^2 a key metric for evaluating the performance of predictive models. Together, these criteria allow for a systematic evaluation of the optimization process, ensuring both accuracy and efficiency in identifying optimal reaction conditions.

All trials converge in 2-3 rounds regardless of the initialization method or aquisition function used. We conclude that the convergence would not be a metric used for comparison in this case.

Acquisition function: EI vs TS

Table 1: R^2 and RMSE results for each Acquisition Function

Initialization Method	Acquisition Function	RMSE	R^2
K-means	EI	0.06	1.0
K-means	TS	0.24	0.94
PAM	EI	0.03	1.0
PAM	TS	0.22	0.95
Random	EI	0.04	1.0
Random	TS	0.21	0.96

Based on **Table 1**: The RMSE and R^2 values for each trial are summarized in Table 1. The best method, numerically, is highlighted.

The results of this test indicate that Expected Improvement (EI) is the preferred acquisition function for BO in this context due to its superior performance in terms of accuracy (Table 1). EI consistently outperformed TS, achieving R^2 values of 1 across all trials and RMSE values below 0.10. TS trials had R^2 values ranging from 0.94 to 0.96 and RMSE values around 0.20. Thus, we conclude that the EI acquisition function consistently yields higher coefficients of determination (R^2) and lower Root Mean Square Error (RMSE) values compared to Thompson Sampling (TS) across all initialization methods. Numerically, BO using EI and k-medoids resulted in the best-fitting model. Consequently, we will use this method when comparing it to MLR.

Part 3: Multiple Linear Regression (MLR)

We wrote code to mimic the BO trial process using MLR in Python in a ipynb file. This code uses the pandas, statsmodels, and matlibplot packages. Specifically, we use pandas[18] to convert the categorical variables (temperature, solvent volume, solvent, catalyst, reagent) into binary dummy variables. We are using dummy variables as our dataset has discrete unique values for each column. This means that even if the value is numerical, it only exists as a categorical value in our dataset. We used the *OLS* function from statsmodels[19] to create a linear model for the data points.

Initial fitted model: Response variable = YP = Yield Percent;

$$YP = \beta_0 + \beta_1 TP105 + \beta_2 TP120 + \beta_3 SV150 + \beta_4 SV263 + \beta_5 OCCK + \beta_6 OCC + \beta_7 OCK + \beta_8 CC1 + \beta_9 CCN + \beta_{10} COCO + \beta_{11} PC$$

List of dummy variables:

TP105 = Temperature 105; TP120 = Temperature 120; SV150 = Solvent Volume 150;
SV263 = Solvent Volume 263; OCCK = O=C([O-])C(C)(C)C.[K+];
OCC = O=C([O-])C.[Cs+]; OCK = O=C([O-])C.[K+]; CC1 = CC1=CC=C(C)C=C1;
CCN = CCCC#N; COCO = CCCCOC(C)=O;
PC1 = P(C1=CC=CC=C1)(C2CCCCC2)C3=CC=CC=C3

The full datasets with all the yields were fitted to the initial model to test the significance of each variable on the response variable. We only wanted to use dummy variables that are significant to a 5% significance level. SV150, SV263, OCCK, and OCK have p-values greater than 0.05, thus, they are not significant enough to be considered in the fitted model and were removed from the fitted model equation.

Fitted model after removing insignificant variables:

$$YP = \beta_0 + \beta_1 TP105 + \beta_2 TP120 + \beta_3 OCC + \beta_4 CC1 + \beta_5 CCN + \beta_6 COCO + \beta_7 PC1$$

List of dummy variables:

TP105 = Temperature 105; TP120 = Temperature 120; OCC = O=C([O-])C.[Cs+];
CC1 = CC1=CC=C(C)C=C1; CCN = CCCC#N; COCO = CCCCOC(C)=O;
PC1 = P(C1=CC=CC=C1)(C2CCCCC2)C3=CC=CC=C3

As the fitted model has seven independent variables, eight initial data points are necessary for the model as it needs enough degrees of freedom for the model to calculate residuals (the errors of the prediction). For this MLR model, we assumed that each independent variable is independent of one another and, therefore, did not test interactions between variables. We ran the BO trial with the initialization function k-medoids and acquisition function EI once again using a batch size of eight in order to produce comparable results with the MLR trial. We made sure that the initial eight experiments used in the BO trial were the same eight being used in the initialization of the MLR trial.

Once the eight initial points were fitted, we started an iterative process. This process is similar to the loop used in BO. It first creates and fits the linear model to all the data points it currently has. The model then predicts eight new points that will result in the maximum yield. These eight points are then queried on the full dataset, where it has the actual yields, and added to the dataset used to train the model. After, analysis plots (linear regression, QQ normal plot of residuals, fitted residual plot) are printed along with the model fit summary and RMSE. We iterated this process until the max yield stayed the same for 5 rounds leading us to conclude that the max yield has been found. In the end, a convergence plot and a plot that mapped the value of RMSE and R^2 were printed.

Results

Bayesian Optimization Result:

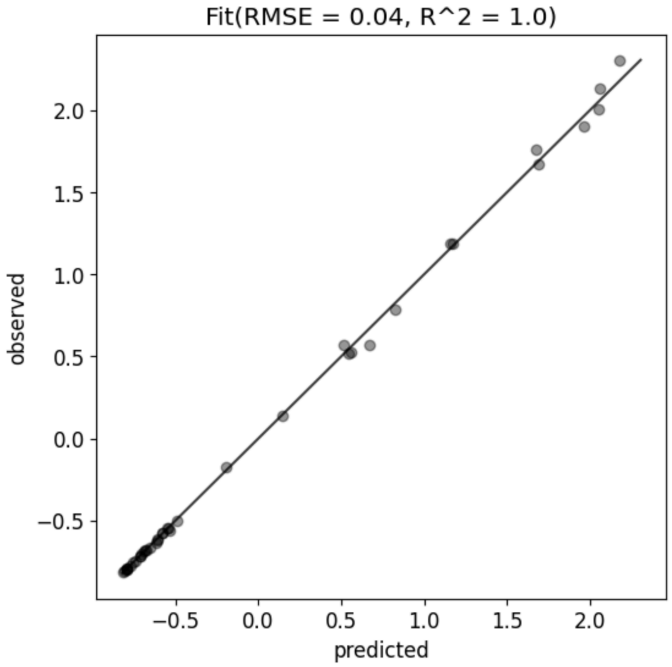


Figure 4: Regression Model Fit (Acquisition Function: EI, Initialization Method k-medoid). Illustrates the predictive performance of the regression model.

The observed values very closely match the predicted values which can be seen by the placements of the points (Figure 4). The RMSE (0.04) indicates minimal error between the observed and predicted values, and the R^2 value of 1.0 demonstrates a perfect fit, confirming the reliability of the model in reflecting the underlying response behaviour.

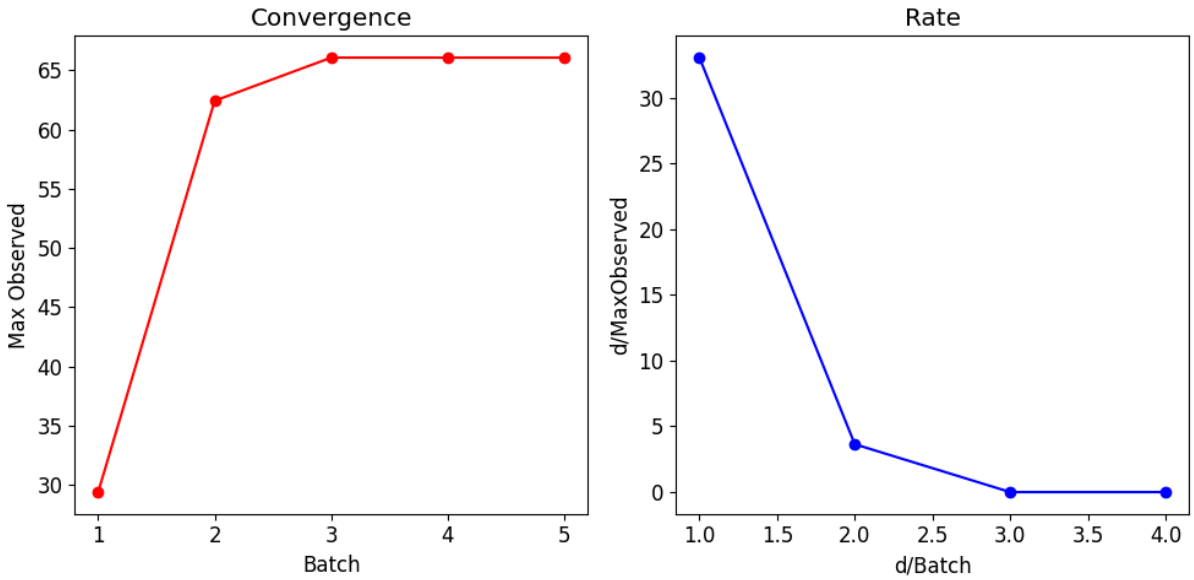


Figure 5: Convergence and Rate of Convergence. (Convergence, a): This plot shows the maximum observed yield versus optimization rounds (batches). (Rate of Convergence, b): The rate of convergence, represented by the first derivative of the maximum observed yield,

The yield converges quickly, with significant improvements in the first two batches and stabilization after the third batch. This shows that optimal yields can be achieved with minimal experimental iterations (Figure 5a). The rate of convergence decreases sharply after the first d/batch and approaches zero by the fourth d/batch. This trend demonstrates rapid initial improvements, followed by diminishing returns as optimization progresses (Figure 5b). The figures collectively show that BO is highly efficient, achieving rapid convergence to the optimal yield with a minimal number of experiments.

Multiple Linear Regression Result:
After running the MLR process for four rounds, the R^2 is 0.8212433001356043 ≈ 0.82 and the RMSE is 9.9708.

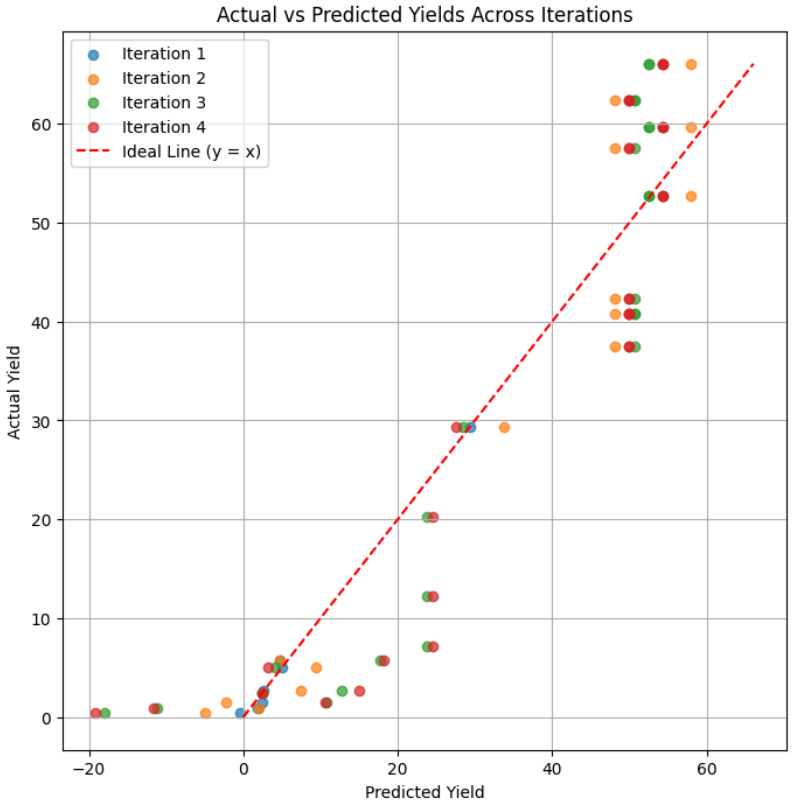


Figure 6: Actual vs Predicted Yields Across Iterations. Compares the predicted yields from the regression model and the actual observed yields over four rounds of optimization. Each iteration is represented by a different colour. The red dashed line represents the line of perfect fit (y = x) when the predicted values are in perfect agreement with the actual values.

From observing Figure 6, the location of the points along the perfect line indicates a fairly decent fit. The predicted values are further from the actual value when the actual values are smaller. The fit is fairly good when the actual value is around 20 or larger. The model under-predicts when the actual value is very close to 0 and around 55 as well. The model over-predicts when the actual value is around the 1 to 13 range and the 37 to 43 range. It tends to under or over-predict at different parts of the dataset but the predictions are still fairly close to the perfect line. The plot reveals outliers in the predicted yield ranges of 0-25 (underestimation) and 42-60 (overestimation), along with negative predicted values. The R^2 value of 0.82 means that the model represents 82% of the variation of the data and is a moderately good fit. An RMSE value of 9.97 means that the predictions are moderately accurate.

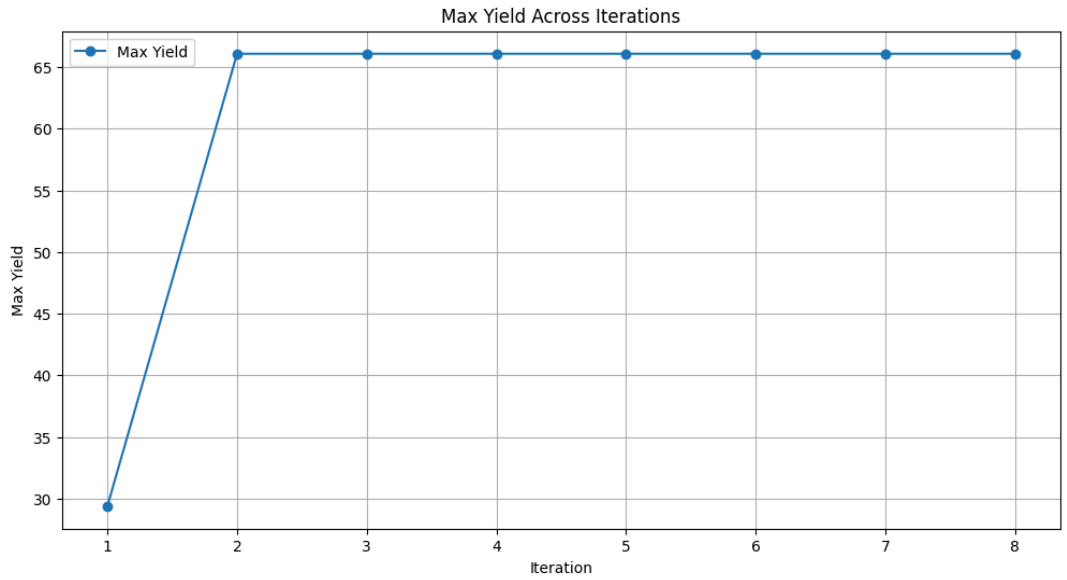


Figure 7: Maximum Yield Progression Over Iterations. Shows the evolution of the maximum observed yield over eight iterations.

The yield increases significantly during the first two iterations and quickly stabilizes at a maximum value of around 67 (Figure 7). This rapid convergence demonstrates the effectiveness of the optimization process in determining the optimal reaction conditions.

Discussion

Comparing the Results:

The results obtained using BO and MLR demonstrate the superiority of BO in optimizing reaction yields in organic synthesis. The comparative analysis, supported by numerical and graphical data, emphasizes both the advantages and disadvantages of each method.

Bayesian Optimization Numerical Results Analysis. The RMSE values for BO (0.04) indicate a near-perfect alignment of predicted and observed yields, with errors amounting to only 0.04%. An R^2 value of 1.0 across all trials using EI confirms that BO captures the underlying patterns in the data with exceptional accuracy.

Overview of Bayesian Optimization. BO is a highly efficient tool for optimizing reaction yields, achieving rapid convergence with significantly fewer experiments (14–21) compared to traditional methods. Its flexibility stems from leveraging acquisition functions like Expected Improvement (EI) and Thompson Sampling (TS), which dynamically balance exploration and exploitation to adapt to the data's characteristics. BO also demonstrates robustness, using a probabilistic framework to handle uncertainty and nonlinear relationships, outperforming simpler models like MLR in accuracy and consistency. However, BO's computational complexity demands greater resources and expertise, and its reliance on surrogate models, such as Gaussian Processes (GP), may limit generalizability to all reaction systems. Despite these challenges, BO offers a transformative and adaptable approach to reaction optimization.

Multiple Linear Regression Numerical Results Analysis. The RMSE of 9.97 demonstrates a significant margin of error in the MLR model's predictions, underestimating yield percentages by 9.97%. An R^2 value of 0.82, although respectable, falls short of BO's performance, indicating less reliability in modelling complex, nonlinear relationships. Through analyzing the model, it is clear that a simple MLR does not accurately represent the distribution of the data set.

Analyzing the Effectiveness of MLR Using Residuals:

Outside of the linear regression and convergence graphs produced during the MLR process, we also produced residual plots to analyze what factors contributed to its shortcomings. The validity of an MLR model heavily relies on the assumption that the residuals are normally distributed, the variance of the residuals is constant and that it follows linearity[9]. We used a QQ normal plot (normality plot) to check if the model follows a normal distribution and a Residuals vs. Fitted Values plot to check if the data has a constant variance.

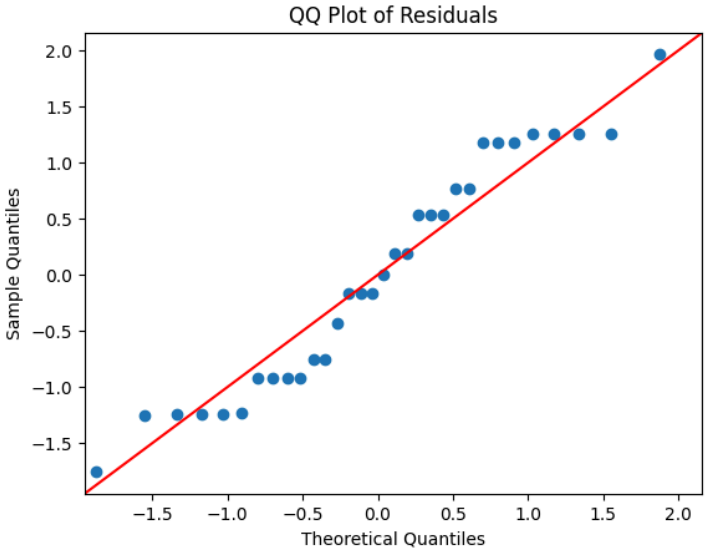


Figure 8: QQ Normal Plot of Residuals. A Quantile-Quantile (QQ) plot is used to compare the distribution of residuals (errors) to a theoretical normal distribution[10].

Figure 8 illustrates the residuals of the model compared to a theoretical normal distribution, with the red diagonal line representing the expected values of normally distributed residuals. It shows deviations from the line, especially at the tails. It appears to be heavy-tailed on both sides which means there are some outliers on the tail ends of the distribution. While some residuals align with the line, deviations indicate that the residuals do not fully follow a normal distribution. Thus, MLR model residuals are not normally distributed, violating one of the key assumptions of linear regression. This suggests potential violations of model assumptions, which may impact the accuracy and reliability of the model.

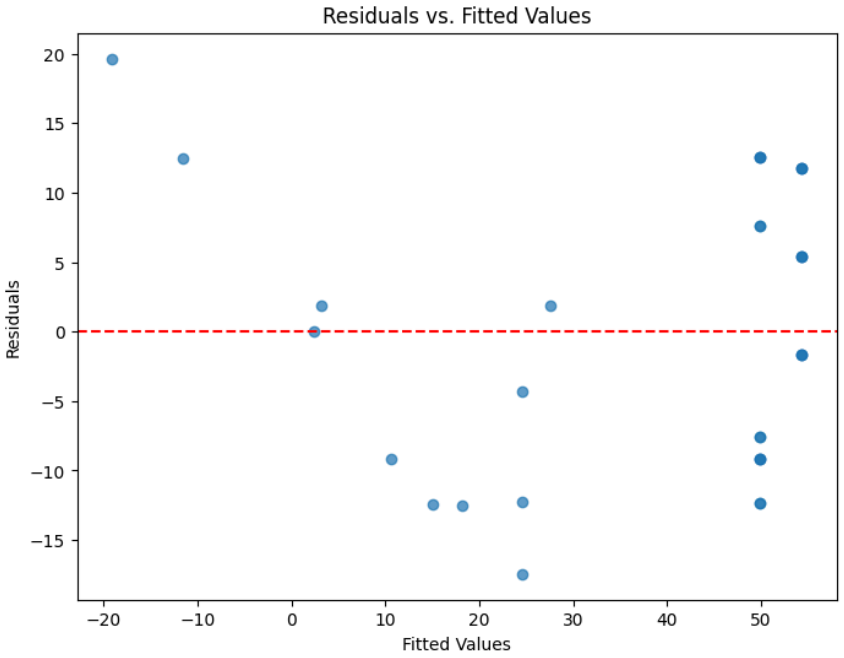


Figure 9: Residual vs Fitted Values. This graph shows the scatter and distribution of the residuals over the fitted values. Ideally, the residuals should have no discernible pattern and should be randomly scattered around zero.

Figure 9 shows the residuals of the MLR model plotted against the fitted values. Ideally, the residuals should be uniformly distributed around zero, indicating a good model fit. However, in this plot, the residuals are widely scattered and unevenly distributed, indicating that the model fails to reflect the data accurately. There seems to be a slight upward-facing curve pattern, which suggests that the residuals follow a non-linear distribution. This non-uniform scatter may indicate that the error depends on the fitted values, which highlights the limitation of MLR when dealing with complex datasets. On the other hand, the graph does not show a funnel pattern, an indication of non-constant variance, we can conclude that the residuals are likely to have a constant variance.

Overview of Multiple Linear Regression. There are two main strengths of using MLR which are its simplicity of usage and interpretation. MLR provides a straightforward framework for understanding the relationship between independent variables and reaction yields. It is easy to implement as it requires fewer computational resources, less data fitting/sorting, and domain expertise compared to BO. However, there are several limitations. One is that it comes with many assumptions. The deviations from such assumptions can result in the model losing validity. Another drawback is that there needs to be at least one more initial experiment compared to the number of independent variables being tested as an extra degree of freedom is needed for the residuals. This limits the amount of different factors and categories you can test as well as the amount of interactions you can test. As MLR is still a linear regression, it struggles with capturing the complex, nonlinear interactions typically seen in chemical optimization. A non-linear model will do a better job of predicting the actual yields compared to a MLR model.

Broader Implications and Future Directions

Our study has several limitations. A significant limitation is the reliance on the Open Reaction Database (ORD) as the primary data source. While comprehensive, the ORD lacks key variables for some reactions, such as dipole moments and has missing yields. To address these gaps, missing values had to be calculated manually, which introduced uncertainties and potential inconsistencies in the dataset. These limitations, along with time constraints and the absence of laboratory resources and robotic automation, further restricted the size and diversity of datasets that could be utilized. This, in turn, narrowed the scope of reaction types that could be studied and optimized in this work.

There are many future opportunities for improving the application of BO in organic synthesis that could be explored. To overcome limitations, future research should focus on conducting full laboratory experiments using robotics, which can automate processes to produce high-quality, consistent datasets while minimizing human error. Expanding datasets to cover a wider range of reaction types and conditions will enhance the validation and applicability of BO across various chemical systems. Detailed sensitivity analyses of BO parameters, such as acquisition functions and batch sizes, will provide critical insights into optimizing the algorithm for specific reactions and experimental setups. Additionally, exploring alternative surrogate models to GP could reduce model dependencies and improve adaptability to more complex or irregular reaction systems. Finally, applying BO to industrially relevant reactions, such as those in pharmaceutical synthesis, will demonstrate its utility in real-world scenarios and help refine its scalability for broader applications.

Conclusions

Our exploration of Bayesian Optimization (BO) in reaction condition optimization highlights its potential as a powerful tool for optimizing organic synthesis reactions. By leveraging the predictive capabilities of Gaussian Process models, BO effectively demonstrates a significant advantage in reducing experimental workloads and refining reaction predictions. Although BO and multiple linear regression (MLR) converge at the same rate, BO consistently outperforms MLR in the accuracy of reaction yield prediction, especially for more complex reactions. This makes BO a better choice for optimizing complex reaction systems. Although we did not directly evaluate the scalability of BO in this study, its ability to handle complex reactions suggests that it is scalable for even larger or more challenging optimization problems. In contrast, MLR, due to its simplicity and computational efficiency, is better suited for less complex reaction systems where the relationships between variables are more linear and simple. These findings highlight the importance of matching the optimization method to the complexity of the problem to be solved. The demonstrated accuracy and adaptability of BO make it an essential tool in organic synthesis research with significant potential for wider application and future advances.

References

(1) Frazier, P. I. Bayesian Optimization. In *Recent Advances in Optimization and Modeling of Contemporary Problems*; Gel, E., Ntaimo, L., Shier, D., Greenberg, H. J., Eds.; INFORMS, 2018; pp 255–278. https://doi.org/10.1287/educ.2018.0188.

(2) Oladipupo, T. Types of Machine Learning Algorithms. In *New Advances in Machine Learning*; Zhang, Y., Ed.; InTech, 2010. https://doi.org/10.5772/9385.

(3) 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. https://doi.org/10.1038/s41586-021-03213-y.

(4) Berk, J.; Gupta, S.; Rana, S.; Venkatesh, S. Randomised Gaussian Process Upper Confidence Bound for Bayesian Optimisation. arXiv June 8, 2020. https://doi.org/10.48550/arXiv.2006.04296.

(5) Jacob Gardner; Geoff Pleiss; Kilian Q. Weinberger; David Bindel; Andrew G. Wilson. GPyTorch: Blackbox Matrix–Matrix Gaussian Process Inference with GPU Acceleration. **2018**.

(6) Mockus, J. On the Bayes Methods for Seeking the Extremal Point. *IFAC Proceedings Volumes* **1975**, *8* (1), 428–431. https://doi.org/10.1016/S1474-6670(17)67769-3.

(7) Hernández-Lobato, J. M.; Requeima, J.; Pyzer-Knapp, E. O.; Aspuru-Guzik, A. Parallel and Distributed Thompson Sampling for Large-Scale Accelerated Exploration of Chemical Space. arXiv June 6, 2017. https://doi.org/10.48550/arXiv.1706.01825.

(8) Benjamin Shields. Edbo. https://github.com/b-shields/edbo (accessed 2024-12-01).

(9) A.J Dobson. *An Introduction to Generalized Linear Models*; Chapman and Hall, 1999.

(10) N.R Draper; H Smith. *Applied Regression Analysis*, 2nd ed.; Wiley, 1981.(10)\uc0\u160{}P McCullagh; J.A Nelder. {\i{}Generalized Linear Models}, 2nd ed.; Chapman and Hall, 1989.

(12) Steven Kearnes. Open Reaction Database. https://github.com/open-reaction-database/ord-data/blob/main/data/0c/ord_dataset-0c75d67751634f0594b24b9f498b77c2.pb.gz (accessed 2024-11-05).

(13) Joo, J. M.; Touré, B. B.; Sames, D. C−H Bonds as Ubiquitous Functionality: A General Approach to Complex Arylated Imidazoles via Regioselective Sequential Arylation of All Three C−H Bonds and Regioselective *N* -Alkylation Enabled by SEM-Group Transposition. *J. Org. Chem.* **2010**, *75* (15), 4911–4920. https://doi.org/10.1021/jo100727j.

(14) Chris Piech. *K Means*. https://stanford.edu/~cpiech/cs221/handouts/kmeans.html (accessed 2024-11-29).

(15) rveerendra400. *Difference between K means and K medoids Clustering*. https://www.geeksforgeeks.org/k-means-vs-k-medoids-clustering (accessed 2024-12-01).

(16) MathWorks. *kmedoids*. kmedoids. https://www.mathworks.com/help/stats/kmedoids.html (accessed 2024-11-29).

(17) Chicco, D.; Warrens, M. J.; Jurman, G. The Coefficient of Determination R-Squared Is More Informative than SMAPE, MAE, MAPE, MSE and RMSE in Regression Analysis Evaluation. *PeerJ Computer Science* **2021**, *7*, e623. https://doi.org/10.7717/peerj-cs.623.

(18) Pandas Development Team. *pandas.get_dummies*. pandas.get_dummies: Pandas Documentation. https://pandas.pydata.org/docs/reference/api/pandas.get_dummies.html (accessed 2024-12-05).

(19) Statsmodels Developers. *Linear Regression*. Statsmodels 0.14.4: Linear Regression. https://www.statsmodels.org/stable/regression.html (accessed 2024-12-05).