Chemically-informed data-driven optimization (ChIDDO): Leveraging physical models and Bayesian learning to accelerate chemical research

Daniel Frey, Xinshu Shang, Juhee Shin, Miguel Modestino

Abstract

Edisonian search approaches are widely used in the chemical sciences to discover reactions, process conditions, material compositions, or product formulations with optimal performance for their intended application. These experimental design methods rely on the generation of grids of variables where experimentally accessible conditions are systematically varied. While these methods are simple to implement, they can result in the oversampling of suboptimal conditions and in information gaps between the selected design parameters. These shortcomings represent significant impediments for expensive experimental campaigns (*e.g.*, during process scale-up, in fine chemicals or pharmaceuticals) or those with large design spaces that can only afford the implementation of coarse experimental grids, underscoring the need for more efficient experimental optimization methods. Here, we introduce a chemically-informed data-driven optimization (ChIDDO) approach, which is a type of multi-information source optimization (MISO), where inexpensive and low-fidelity information obtained from physical models of chemical processes are combined with expensive and high-fidelity experimental data to optimize a common objective function. While MISO algorithms have been previously implemented to improve BO in computational problems, the implementation of ChIDDO can extend these advantages to chemical experimentation. We will show how the ChIDDO algorithm compares to traditional BO approaches using benchmark objective functions. We will then show how ChIDDO can be used to optimize an electrochemical reaction engineering problem more efficiently than BO.

Introduction

Edisonian search approaches are widely used in the chemical sciences to discover reactions, process conditions, material compositions, or product formulations with optimal performance for their intended application. These experimental design methods rely on the generation of grids of variables where experimentally accessible conditions are systematically and/or combinatorically varied. While these methods are simple to implement, they can result in the oversampling of suboptimal conditions and in information gaps between the selected design parameters, slowing and sometimes preventing the identification of optimal conditions.[1] These shortcomings represent significant impediments for expensive experimental campaigns (*e.g.*, during process scale-up, in fine chemicals or pharmaceuticals) or those with large design spaces that can only afford the implementation of coarse experimental grids, underscoring the need for more efficient experimental optimization methods[2].

To accelerate experimental optimization, Bayesian optimization (BO) has been widely implemented in many different fields of research[3-9]. BO uses Bayesian statistics to generate predictions of the design space [surrogate models (SMs)] and then interprets the SMs using Bayesian reasoning with the purpose of selecting the most informative experiments[10]. In recent years, BO has been implemented for various applications in the chemical sciences including materials discovery and prediction of their properties[11-21], design of reactors and chemical processes[22-33], and the optimization of energy storage materials and devices[34-38]. Data-driven optimization methods such as BO learn and evolve with new experimental data, but they lack *a priori* knowledge of the physical laws that dictate the behavior of the chemical system under study. This can result in the need for large experimental campaigns to accurately optimize the design space. On the other hand, optimization methods based on physical models (*e.g.*, density functional theory, molecular dynamics, continuum models, etc.) could be used to identify optimums without the need to perform experimental searches, but they often lack the accuracy to effectively capture the complexity of real systems or require inaccessibly-large computational power. Given the advantages and shortcomings of both optimization approaches, there is an opportunity to leverage *a priori* chemical knowledge in data-driven optimization to reduce the data needs and allow for faster identification of optimum.

Herein, we introduce a chemically-informed data-driven optimization (ChIDDO) approach, which is a type of multi-information source optimization (MISO), where inexpensive and low-fidelity information obtained from physical models is combined with expensive and high-fidelity experimental data to optimize a common objective function. While MISO algorithms have been previously implemented to improve BO in computational problems[39-41], the implementation of ChIDDO can extend these advantages to chemical experimentation. In this study we develop a generalizable ChIDDO framework that sources *a priori* knowledge from physical models to accelerate BO algorithms.

Methods

BO Algorithm Description

BO algorithms consist of two main components: a SM and an acquisition function. The SM is used to predict the value of the cost function, y^{pred}, for any set of conditions, $\boldsymbol{x_i}$, in the design space bounded by $\boldsymbol{x_{LB}}$ and $\boldsymbol{x_{UB}}$, which are arrays of the same dimensionality as the design space, n_d. The SM is trained using N^{exp} experimental evaluations of the cost function, $\boldsymbol{y}^{exp}$ at $\boldsymbol{x}^{exp}$. $\boldsymbol{x}^{exp}$ is a matrix with N^{exp} rows and n_d columns, indicating the locations of each of the known design conditions for each experiment. $\boldsymbol{y}^{exp}$ is an array of N^{exp} evaluations of the cost function at each condition in $\boldsymbol{x}^{exp}$. In this study, we use a gaussian process regressor (GPR) with a radial basis function (RBF) kernel plus white noise as the SM.

An acquisition function is used to select the next design condition(s) to evaluate, $\boldsymbol{x}^{next}$, based on how informative the design conditions will be in the goal of optimizing the cost function. Many different acquisition functions for BO have been developed, and two of the most common are Probability of Improvement (PI) and Upper confidence bound (UCB). In addition to these, we have developed a modified ranked-batch (MRB) mode sampling function inspired by the work of Cardoso et al.[42]. The equations for each of acquisition functions are provided in the Supplemental Information. The acquisition function uses the current information, $\boldsymbol{x}^{exp}$ and $\boldsymbol{y}^{exp}$, and the SM predictions to calculate how informative a possible design condition, $\boldsymbol{x}^{poss}$, is expected to be be based on the criteria in the respective acquisition function. The

algorithm then uses a gradient descent method to find a local minimum of the acquisition function, and assigns it to $\boldsymbol{x}^{next}$. Depending on the batch size of subsequent experiments in each ChIDDO step, n_b, multiple design conditions can be added to $\boldsymbol{x}^{next}$ by repeating this acquisition function minimization step. After $\boldsymbol{x}^{next}$ is selected, the objective function value(s) are determined experimentally to obtain $\boldsymbol{y}^{next}$. Subsequently, $\boldsymbol{x}^{next}$ and $\boldsymbol{y}^{next}$ are appended to $\boldsymbol{x}^{exp}$ and $\boldsymbol{y}^{exp}$.

The PI and UCB algorithms were run based on their implementation in the modAL active learning framework, which is described in the Supplemental Information. The general framework for the BO algorithms presented was also based on the modAL framework. The MRB acquisition function calculated a score consisting of three normalized parameters: a distance score, Δ, an uncertainty score, Γ, and an the objective function prediction, Ω. These parameters are calculated by:

$$\Delta_i = 1 - 1/\left(1 + min(\|x_i - x^{exp}\|_2)\right) \quad (1)$$

$$\Gamma_i = \sigma_i \quad (2)$$

$$\Omega_i = y_i^{pred} \quad (3)$$

where $\min(\|\boldsymbol{x_i} - \boldsymbol{x^{exp}}\|_2)$ is the minimum distance from the set of distances calculated between the current design point in the gradient descent process, $\boldsymbol{x_i}$, and each of the known points, $\boldsymbol{x}^{exp}$. σ_i is the standard deviation of the GPR prediction at $\boldsymbol{x_i}$, and y_i^{pred} is the prediction by the GPR at $\boldsymbol{x_i}$. The score that is calculated at each step in the gradient descent process is:

$$Score_j = \delta_j \Delta_i + \gamma_j \Gamma_i + \omega_j \Omega_i \quad (4)$$

where j is the batch number of the algorithm and δ_j, γ_j, and ω_j are parameters that change as more batches are completed to weight the components differently based on exploration or exploitation. These parameters are calculated by:

$$\delta_0 = 1 - N_{init}^{exp}/N \quad (5)$$

$$\gamma_0 = (1 - \delta_0)/2 \quad (6)$$

$$\omega_0 = (1 - \delta_0)/2 \quad (7)$$

$$\delta_j = \delta_0 \varepsilon_j \quad (8)$$

$$\gamma_j = \gamma_0 + \left[(\delta_0 - \delta_j)\rho\right] \tag{9}$$

$$\omega_j = \omega_0 + \left[(\delta_0 - \delta_j)(1-\rho)\right] \tag{10}$$

$$\varepsilon_0 = \kappa^{1/N_b} \tag{11}$$

$$\varepsilon_j = \varepsilon_0^j \tag{12}$$

where N_{init}^{exp} is the initial number of experimental points, N is the total number of experiments to be run, κ is a parameter that determines the rate of the change from exploration to exploitation, ρ is the proportion of change to the parameters γ_j and ω_j, and N_b is the total number of batches that take place before N is reached.

To initiate the algorithm, N_{init}^{exp} of evenly distributed random points were chosen as the initial set of experiments. For the BO algorithm without use of a physics model (referred to as BO from this point on), only these initial points, $\boldsymbol{x}_{\boldsymbol{init}}^{\boldsymbol{exp}}$, were passed to the SM. After each batch of BO, ($\boldsymbol{x}^{\boldsymbol{exp}}$, $\boldsymbol{y}^{\boldsymbol{exp}}$) increases in size by the batch size, n_b. For the ChIDDO algorithm, before ($\boldsymbol{x}^{\boldsymbol{exp}}$, $\boldsymbol{y}^{\boldsymbol{exp}}$) is passed to the SM, a number equal to $\left(N - size(\boldsymbol{x}^{\boldsymbol{exp}})\right)$ of evenly distributed random points from the physics model ($\boldsymbol{x}^{\boldsymbol{phys}}$,$\boldsymbol{y}^{\boldsymbol{phys}}$) is appended to ($\boldsymbol{x}^{\boldsymbol{exp}}$, $\boldsymbol{y}^{\boldsymbol{exp}}$). The SM then predicts $\boldsymbol{y}_{\boldsymbol{i}}^{\boldsymbol{pred}}$ based on the combination of experimental and physics-model predicted conditions. A general process diagram of this algorithm is shown in Figure 1.

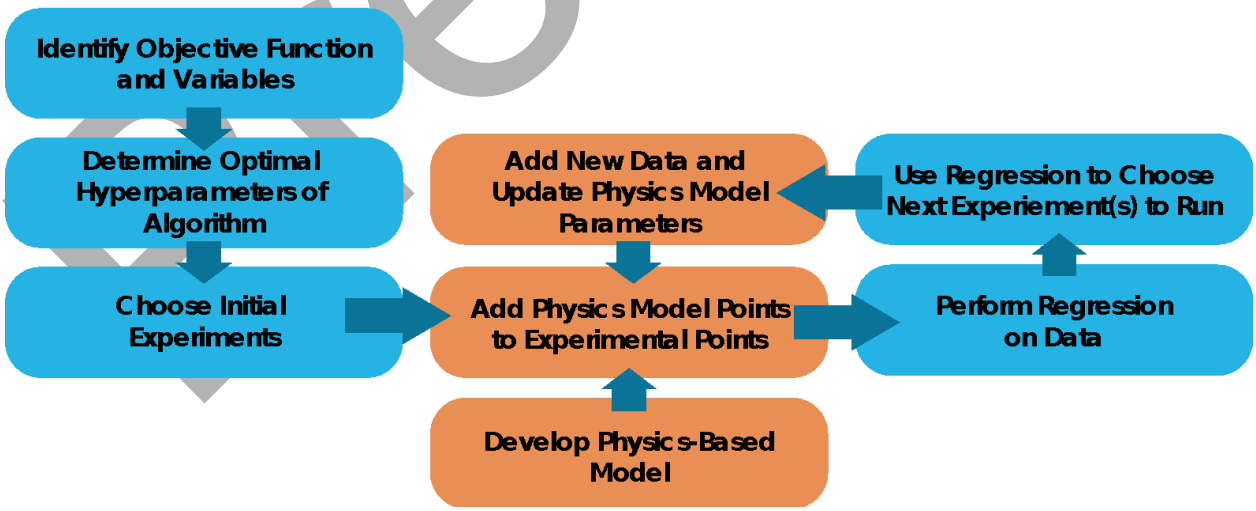

Figure 1. Process diagram of the BOphys algorithm. The orange blocks correspond to the added steps to include the consideration of the physics model. The blue blocks correspond to the steps to run BO.

Updating the physics model parameters in ChIDDO

Each objective function was defined by a set of base parameters that could be modified to generate different objective functions with a similar shape. As the *a priori* guess for each objective function in the ChIDDO algorithm, a set of base case parameters was used, which are provided in the Supplemental Information. Since the initial physics model parameters are only a guess, the parameters were updated after each batch of experiments to improve the physics model guess based on the experimental observations. The parameters were updated by using a gradient descent method to fit the parameters of the objective function with the known points, $(\boldsymbol{x^{exp}}, \boldsymbol{y^{exp}})$. The updated model parameters were then used to calculate ($\boldsymbol{x^{phys}}, \boldsymbol{y^{phys}}$) for the following batch.

Benchmark Objective Functions

Common objective functions for optimization benchmarking were selected and are described further in the Supporting Information. Each objective function design space was defined by lower bounds and upper bounds. In addition to testing 2-dimensional design spaces, we studied 3-,4- and 6-dimensional spaces. Unless otherwise specified, the values calculated as experimental values, y^{exp}, were exactly equal to the objective function calculation, given the set of parameter values. Under conditions when noise was added, the objective function values were calculated as:

$$y_i^{noise} = y_i^{exp} + [(y_{max} - y_{min})(2\, rand(O,1) - 1)]n \qquad (13)$$

where y_{max} is the maximum value of the objective function, y_{min} is the minimum value of the objective function, and n is the noise level, defined as the maximum allowable value that could be added or subtracted from y_i^{exp}, which can be viewed as a percentage of the range of y. The n values that were tested were 0.025, 0.05, and 0.1.

Hyperparameter Optimization

In the BO algorithm described above, there are four hyperparameters that needed to be optimized before testing: N_{init}^{exp}, n_b, ρ, and the exploration-exploitation rate, κ.

For each study of an optimization algorithm on an objective function, the hyperparameters of the algorithm were first optimized on 20 different alternate models. Alternate models are instances of the objective function of interest with different model parameters. The hyperparameters were optimized using a BO algorithm to maximize the following function:

$$Val_{hyper} = \sum_i \sum_j y_{max,j}^{exp} * \left(1 - j/(N)\right) \quad (14)$$

where *i* represents the different alternate models, *j* represents the experiment number in the range (1,*N*), and $y_{max,j}^{exp}$ is the maximum value from the first *j* values of y^{exp}. Equation 14 was formulated assuming the ideal performance of a BO algorithm would be finding the maximum valued solution in as few experiments as possible. After the BO algorithm selects and tests 25 hyperparameter combinations, the combination with the highest score, max(Val_{hyper}), was chosen for further testing. The chosen hyperparameter combination was used in subsequent tests in which one of the alternate models was studied using different selections of the initial conditions. This was done to help understand the robustness of the method to variations in the starting conditions.

Baseline search algorithms

Two different baseline search methods were tested: a grid search and a random search. 100 trials were done for each search method and *N* experimental points were selected for each trial. For the grid search, equally spaced points were selected sequentially until *N* experimental points were selected. A new starting location was chosen for each trial. For the random search, conditions were chosen at random from the uniform design space defined by the upper and lower bounds. This random search did not consider the locations of the previous selections, so it was possible to have poor representation of the design space (*i.e.*, clustering of points).

Simplified physics model

In order to study how the algorithm performs when the physics model is not an accurate representation of the system under study, a single objective function was used as the physics model to optimize a combined objective function. The values of the combined objective function were calculated as:

$$y_i^{mixed} = r * y_i^1 + (1 - r) * y_i^2 \quad (15)$$

where r is the mixing ratio, y_i^1 is the value of the first objective function, and y_i^2 is the value of the second objective function. For example, the Rosenbrock function could be added to the Sphere function with an r of 0.9. In this case, the objective function would more closely resemble the Rosenbrock function.

Results

BO enhancements over Edisonian approach

To demonstrate the advantages of implementing a BO strategy over an Edisonian approach, we studied the performance of the different optimization approaches on common benchmark functions. Each of these benchmark functions has a different shapes and optimization complexity, and by running the algorithms on these different objective function, we attempted to gain insight into the behavior of the different algorithms studied. For conciseness, here we present the results for various optimization runs using a Sphere function (Figure 2). A full list of the objective functions, their equations, the base parameters, and optimization results can be found in the Supplemental Information.

In our framework, we consider experimental sets, $\boldsymbol{S}$, which consist of N^{exp} number of experiments with conditions $\boldsymbol{x}^{exp}$ resulting in output performance, $\boldsymbol{y}^{exp}$. The purpose of the BO algorithm is to identify the set of experiments, $\boldsymbol{S_{BO}}$, with the lowest number of experiments, N_{BO}, that leads to the optimal conditions, $\boldsymbol{x_{opt}}$, that maximize the value of $\boldsymbol{y_{max}^{exp}}$. We then contrast $\boldsymbol{S_{BO}}$ to Edisonian experimental sets, $\boldsymbol{S_{ed}}$, that follow either a grid or a random search approach, and we evaluate two performance metrics: the normalized deviation from the optimum value, d_y, and the minimum distance from the optimum, d_x, identified by each set of experiments. These two quantities are calculated as,

$$\mathrm{d_y} = \min \frac{y_{max} - y_i^{exp}}{y_{max} - y_{min}} \quad (16)$$

$$d_x = \min \left| \boldsymbol{x_i} - \boldsymbol{x_{opt}} \right| \quad (17)$$

where y_{max} is the maximum possible value of the cost function within the constraints of the experimental parameters and y_{min} is the minimum possible value of the cost function within the constraints of the experimental parameters.

In Figure 2, d_y is plotted against the number of experiments, *N*, comparing the Edisonian methods with BO and ChIDDO. The plots for d_x can be found in the Supplemental Information. Figure 2A shows how the different search algorithms compare using the 2D Sphere objective function. Even for this simple, parabolic function, the systematic grid search and random search underperform comparatively to BO or ChIDDO. d_y after 20 experiments, d_{y20}, were 4 and 3 x 10^{-2} for the grid and random search algorithms, respectively. In comparison, d_{y20} for BO and ChIDDO were 6 and 7 x 10^{-3}, respectively. As the design space moves to higher dimensions, Figures 2B and 2C show that the differences between the algorithms increase with dimension size. For the 3D Sphere objective function the enhancements are more drastic with d_{y20} being two orders of magnitude smaller for the BO algorithms compared to the Edisonian algorithms. Because of the larger design space to sample, the grid and random search methods are not capable of searching a fine enough space to find values close to the optimal. It is of interest that the d_{y20} for BO and ChIDDO were very similar, possibly the Sphere objective function has only one optimum and is therefore easy to solve for any optimization algorithm. As the number of dimensions increases, ChIDDO tends to find near optimal values with fewer experiments than BO. This enhancement is likely due to the fact that ChIDDO relies on the physics model to help more rapidly locate optimal conditions.

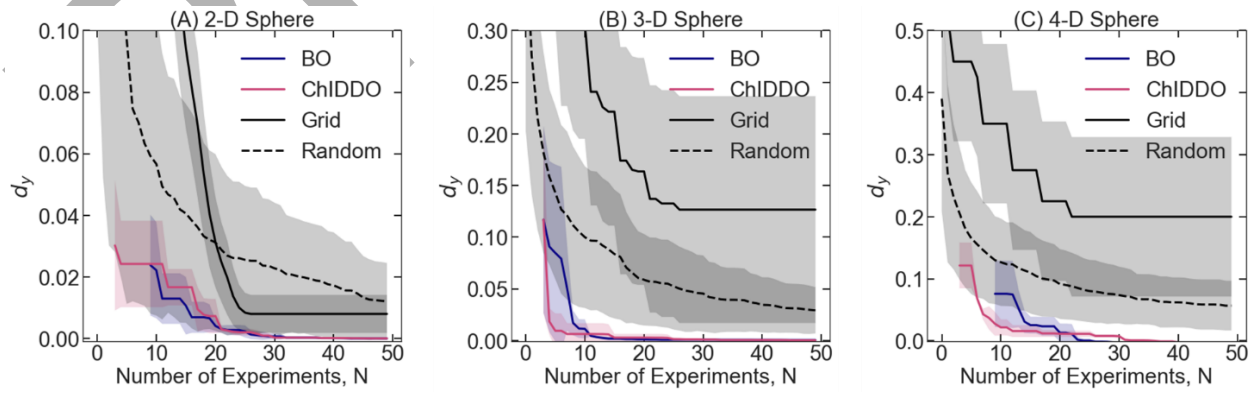

Figure 2. d_y versus number of experiments, *N*, comparing BO and ChIDDO with the Edisonian random and grid search. (A) 2D Sphere, (B) 3D Sphere, (C) 4D Sphere. For each curve, 20 separate searches, S, were performed, and the average of the results are the lines shown. The shadow around each of the lines represents the standard deviation. For

each of the BO/ChIDDO experiments, the MRB acquisition function was used. The objective function parameter information is provided in the Supplemental Information.

Comparison of different acquisition functions

To study how the different acquisition functions performed, they were compared to one another using different objective functions. In order to compare how the different acquisition functions behave at different dimensions, a single objective function class (Hartmann) is shown in the main text with 3, 4, and 6 dimension instances. The graphs for this comparison on other objective functions is provided in Supplemental Information. The Hartmann objective function class was chosen due to its more complex structure (i.e. multiple local optima) and because different dimensions can be compared. Figure 3 shows the comparison of the three different acquisition functions used in the BO algorithm tested on the Hartmann objective functions. Figure 3A shows the comparison on the 3D Hartmann function. The d_{y20} values for MRB, PI, and UCB were 0.245, 0.122, and 0.259, respectively. After the 20th experiment, d_y for both PI and MRB decreases to close to zero, while the d_y for UCB remains near 0.25. Since there are multiple local optima present in this objective function, this result could indicate that the UCB acquisition function found one of these local optima. This is further corroborated when looking at the d_x versus N graph shown in Figure S41 in the Supplemental Information, which shows that PI and MRB have comparable behavior in locating the optimal conditions, while UCB shows a decreased performance. Figure 3B shows the comparison on the 4D Hartmann function. Interestingly, all the acquisition functions perform similarly with $d_{y,20}$ values on the order of $2x10^{-2}$. In addition, the d_y for all the acquisition functions decrease towards zero. The comparison on the 6D Hartmann function is shown in Figure 3C. In this case, PI clearly performs the best with a $d_{y,20}$ of 0.254, compared with that of MRB (0.478) and UCB (0.362). It is interesting to note that the standard deviations for the 6D graphs are much larger than for the smaller dimensions. This indicates that the different random starting conditions affected the d_y values more for the 6D space compared with the 3D and 4D spaces. This is most likely due to the larger size of the 6D space making it more difficult to locate the global optimum point.

These acquisition functions were also studied and compared using the ChIDDO algorithm in Figure 4. For the comparison on the 3D Hartmann function shown in Figure 4A, MRB had an order of magnitude lower d_{y20} value (0.026) than the other acquisition functions (PI=0.249, UCB=0.252). Interestingly, a different trend is seen for the 4D Hartmann function in Figure 4B, in which the UCB acquisition function has much lower d_y values. After the 20th experiment, the d_y for MRB and PI both decreased closer to zero, but d_y for UCB was still lower. This shift in results could be due to the inaccuracy of the initial physics model for the 4D Hartmann function compared to the 3D function. The three initial parameters of the Hartmann function are (1,1,1), but the parameters for the 4D Hartmann function that was tested were (0.375, 1, 0.86). In contrast, the parameters for the 3D Hartmann function under study were (1.14, 1.03, 0.98). Figure 4C shows the comparison on the 6D Hartmann function, and the result is similar to that of the 3D function. The $d_{y,20}$ values for MRB, PI, and UCB were 0.097, 0.215, and 0.786, respectively. These results indicate that the MRB and PI acquisition functions appear to be more robust, regardless of initial starting position. When comparing the BO results to the ChIDDO results, it appears that the ChIDDO algorithm performs similarly or better for all of the objective functions. These results indicate that our proposed MRB acquisition function combined with ChIDDO outperforms other acquisition functions when applied to complex objective functions with multiple local optima.

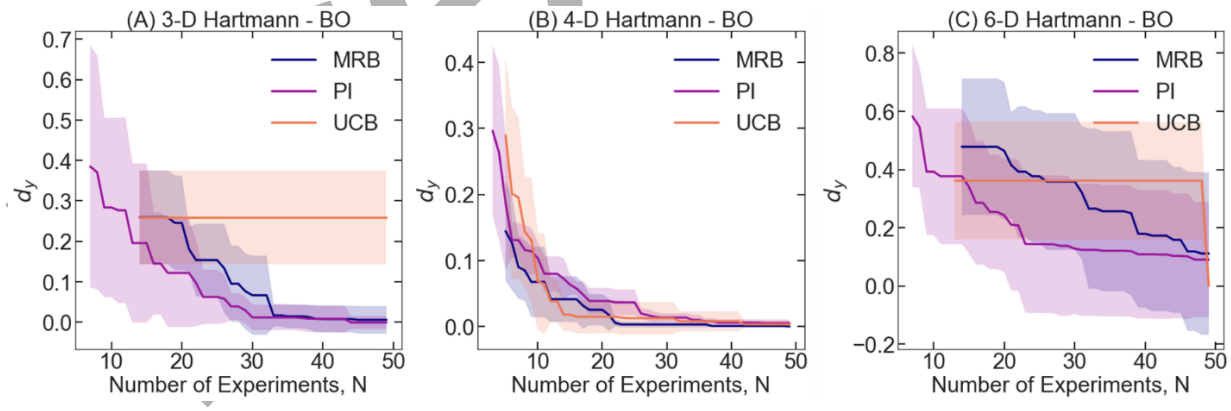

Figure 3. d_y versus number of experiments, N, comparing the MRB, PI, and UCB acquisition functions using BO. (A) 2D Camel, (B) 3D Hartmann, (C) 4D Sphere. For each curve, 20 separate searches, S, were performed, and the average of the results are the lines shown. The shadow around each of the lines represents the standard deviation.

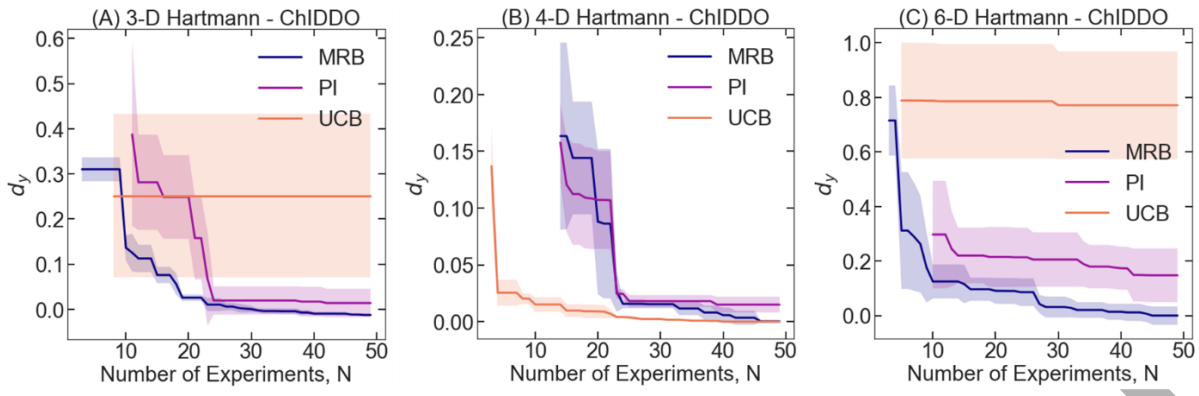

Figure 4. d_y versus number of experiments, *N*, comparing the MRB, PI, and UCB acquisition functions using ChIDDO. (A) 3D Hartmann, (B) 4D Hartmann, (C) 6D Hartmann. For each curve, 20 separate searches with different initial conditions, *S*, were performed, and the average of the results are the lines shown. The shadow around each of the lines represents the standard deviation. The parameters used in the objective function are provided in the Supplemental Information.

Quantifying the effect of experimental noise

So far, we have assumed that experiments run under conditions, x_i, result in exact values of the objective function of interest, $f(x_i) = y_i$. However, experimental measurements often possess a significant degree of noise. To quantify the effect of the experimental noise and to determine the robustness of the BO and ChIDDO algorithms to noisy experiments, different levels of random noise were added to the objective functions.

Figure 5 compares d_y for different levels of noise using the BO algorithm with the MRB acquisition function. In Figure 5A, the noise effects on the 3D Hartmann function are shown. The d_y for n=0.1 appears to be similar to the other noise values until about the 20th experiment, and then the d_y decreases very slowly to about 0.1. For the other noise values, the d_y curves appear to behave in similar ways, decreasing until about 33 experiments where d_y approaches 0. The comparison is shown for the 4D Hartmann function in Figure 5B. Here, the d_y for n=0.1 is consistently larger than the other noise levels and levels off around 0.03. For the other noise levels, the d_y values decrease close to zero, with n=0.025 and n=0.05 performing slightly worse than n=0. For the 6D Hartmann function in Figure 5C, the d_y values for n=0.05 are the highest, while d_y for n=0.025 is the lowest. are consistently higher than that of the other lower noise levels.

These results indicate that the BO algorithm is more resistant to noise effects in low-dimensionality design spaces.

The effects of noise on the ChIDDO algorithm were also studied. Overall, the d_y values for ChIDDO were lower than that of BO, especially for smaller numbers of experiments. For the 3D Hartmann function in Figure 6A, the observations are similar to that of BO as the noise had only a small impact on the optimization. For the 4D Hartmann function in Figure 6B, the d_y values for each noise level are similar after the 25th experiment. Prior to this, the d_y values for n=0.025 and n=0.05 are lower than that of no noise level. Comparing this result to Figure 5B, it appears that the use of the ChIDDO algorithm allows the optimization to be more resistant to noisy data at later experiment values when compared to BO. However, the BO algorithm performs slightly better with few experiments. Contrary to this, Figure 6C shows that the ChIDDO algorithm performs much better overall for the 6D Hartmann function compared to BO. Similar to the results of the ChIDDO algorithm on the other functions, the difference in d_y between the noise levels is minimal.

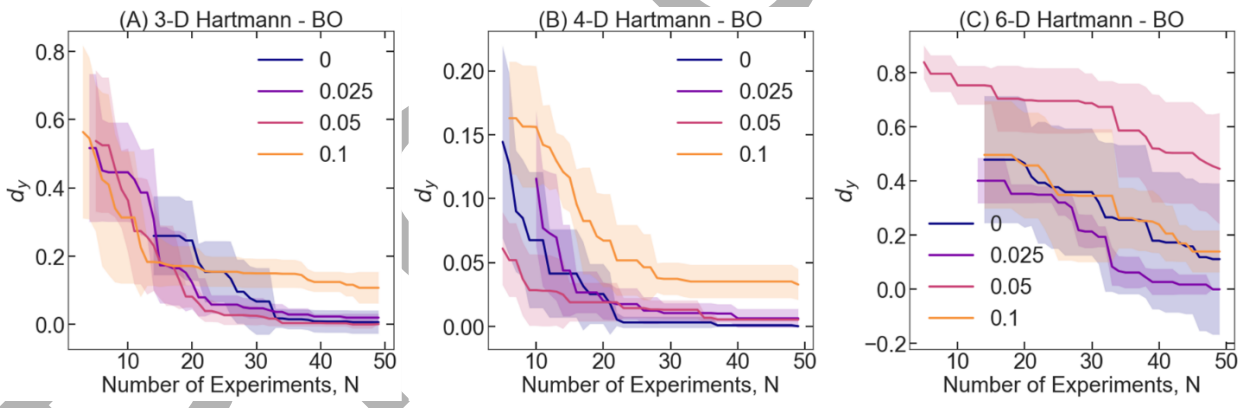

Figure 5. d_y versus number of experiments, N, comparing different noise levels, n, using the BO algorithm. (A) 3D Hartmann, (B) 4D Hartmann, (C) 6D Hartmann. For each curve, 20 separate searches, S, were performed, and the average of the results are the lines shown. The shadow around each of the lines represents the standard deviation. For each of these studies, the MRB acquisition function was used.

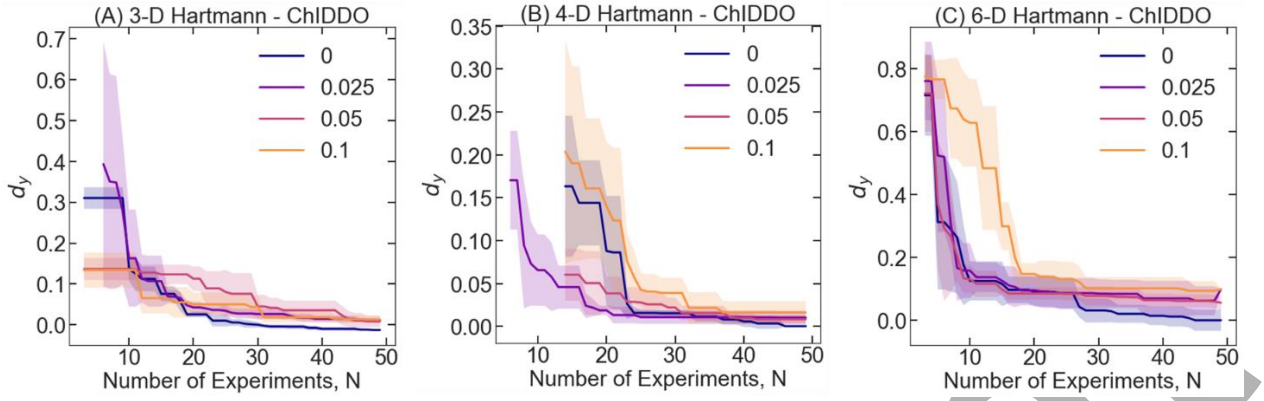

Figure 6. d_y versus number of experiments, N, comparing different noise levels, n, using the ChIDDO algorithm. (A) 3D Hartmann, (B) 4D Hartmann, (C) 6D Hartmann. For each curve, 20 separate searches, S, were performed, and the average of the results are the lines shown. The shadow around each of the lines represents the standard deviation. For each of these studies, the MRB acquisition function was used.

Quantifying effects of physical model accuracy

Experiments in the chemical sciences and engineering are often performed under complex geometries, involve multiple kinetic and transport processes, and require molecular-level descriptions of the species involved. Detailed multidimensional models of these complex chemical systems are often intractable, requiring simplified semi-empirical models that capture with a sufficient level of accuracy the experimental observations. These simplified physical models with various levels of accuracy can still be used in ChIDDO algorithms as they can help guide the optimization and be complemented by the obtained experimental data. To understand how simpler and less accurate models affect the performance of ChIDDO, we attempted to optimize a mixed objective function using a physical model that described only one of the objective functions. The mixed objective function consists of a linear combination of two objective function values as shown in Equation 15. Figure 7 compares different values of r for two different objective function combinations. Figures 7A,C show d_y for the combination of the 2D Rosenbrock and 2D Sphere objective functions. The "physics" model used to guide the algorithm described either the Rosenbrock or the Sphere function. For the example where the Rosenbrock function is used as the physics model, an r of 0.1 indicates that the value of each data point is 10% of the 2D Rosenbrock value plus 90% of the 2D Sphere value. Therefore, a low r indicates that the physics model and the experimental points are dissimilar. Figure 7A

shows the comparison of different r values when the Rosenbrock function is used as the "physics" model. Interestingly, an r value of 0.9 indicates that the physics model and objective function combination are very similar, but the d_y for r=0.9 is higher than for the other ratios. Since the Rosenbrock function has a large part of its design space with similar, close to optimal values, the use of this function as a physics model may not help guide the algorithm in a significant way. In Figure 7C, the 2D Sphere and 2D Rosenbrock objective function mixed is compared using the Sphere function as the "physics" model. The d_y for r=0.9 is the lowest, reaching a value close to zero at the 20th experiment. The other r values perform similar to each other and the d_y values are only slightly higher than for r=0.9. Figures 7B,D show a parallel comparison using the 4D analogs of the objective functions in Figures 7A,C. In Figure 7B, the 4D objective function combination is studied with the 4D Rosenbrock function as the physics model. We observe similar behavior in d_y values between r=0.1 and r=0.9, which both decrease close to zero after the 20th experiment. Interestingly, the d_y for r=0.5 was higher than the other values for all experiments. In Figure 7D, which shows the 4D objective function combination with the 4D Sphere functions as the physics model, the d_y for r=0.9 is much higher than the values for the other ratios. Even with the dissimilar physics model in a 4D design space, the d_y for r=0.9 reaches a value of around 0.02 by the 50th experiment. Based on all of the results in Figure 7, it appears that the ChIDDO algorithm is robust when using physics models that are simple or incorrect.

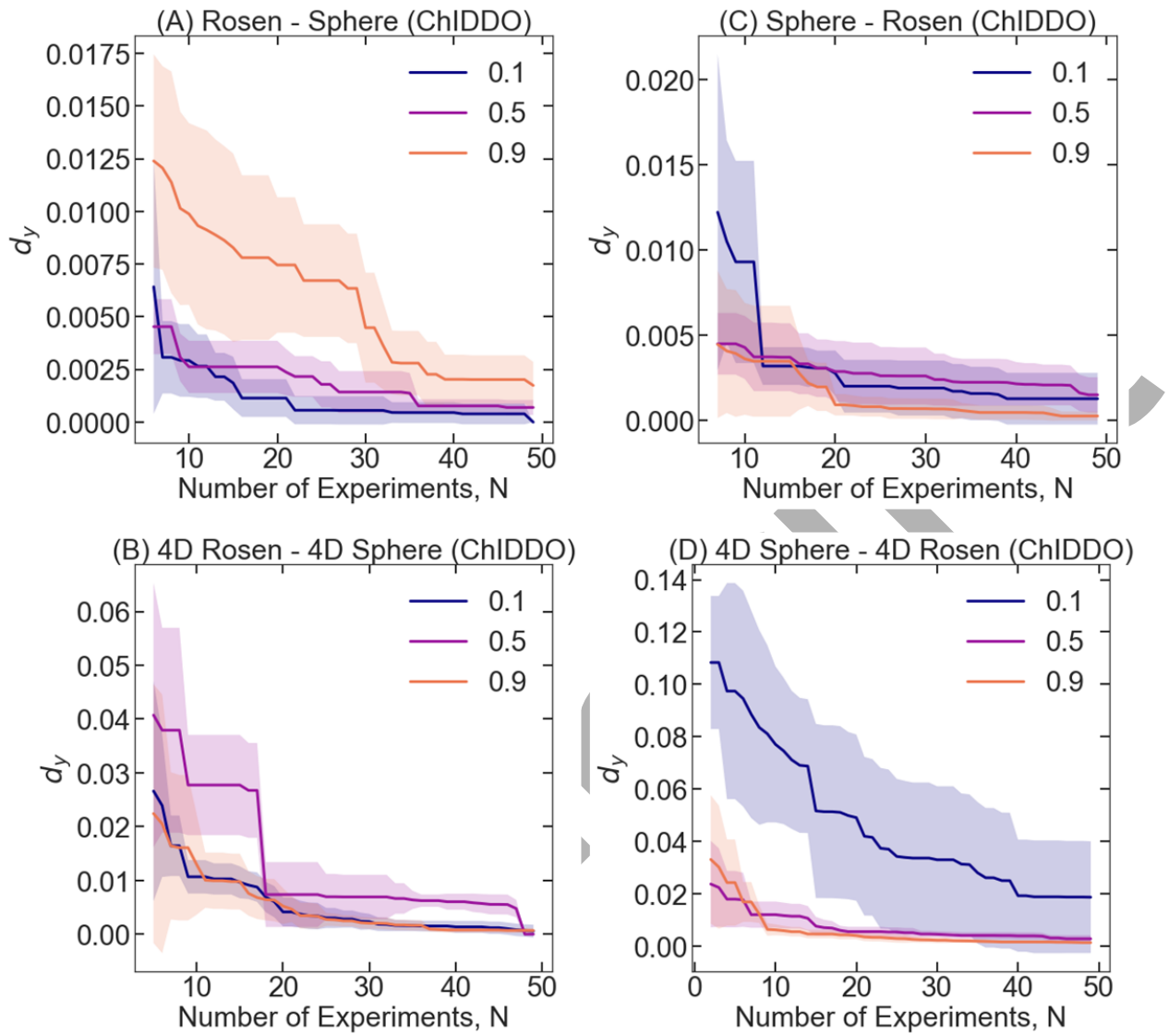

Figure 7. d_y versus number of experiments, N, for different objective function mixing ratios. (A) 2D Rosenbrock mixed with 2D Sphere using Rosenbrock as physics model. (C) 2D Rosenbrock mixed with 2D Sphere using Sphere as the physics model. (B) 4D Rosenbrock mixed with 4D Sphere using Rosenbrock as the physics model. (D) 4D Rosenbrock mixed with 4D Sphere using Sphere as the physics model. For each curve, 20 separate searches, S, were performed, and the average of the results are the lines shown. The shadow around each of the lines represents the standard deviation. For all of these graphs, ChIDDO was used as the AL algorithm and MRB was used as the acquisition function.

Acknowledgments

We thank Prof. Christopher Musco for insightful comments and discussions. We thank NYU Tandon School of Engineering and the National Science Foundation for their generous financial support. This material is based upon work supported by the National Science Foundation under grant no. 1943972.

References

1. Blanco, D. E.; Lee, B.; Modestino, M. A., Optimizing organic electrosynthesis through controlled voltage dosing and artificial intelligence. *Proceedings of the National Academy of Sciences* **2019,** *116* (36), 17683-17689.
2. Grover, A.; Markov, T.; Attia, P.; Jin, N.; Perkins, N.; Cheong, B.; Chen, M.; Yang, Z.; Harris, S.; Chueh, W., Best arm identification in multi-armed bandits with delayed feedback. *arXiv preprint arXiv:1803.10937* **2018**.
3. Xue, D.; Balachandran, P. V.; Hogden, J.; Theiler, J.; Xue, D.; Lookman, T., Accelerated search for materials with targeted properties by adaptive design. *Nat. Commun.* **2016,** *7* (1), 1-9.
4. Vahid, A.; Rana, S.; Gupta, S.; Vellanki, P.; Venkatesh, S.; Dorin, T., New bayesian-optimization-based design of high-strength 7xxx-series alloys from recycled aluminum. *JOM* **2018,** *70* (11), 2704-2709.
5. Li, C.; de Celis Leal, D. R.; Rana, S.; Gupta, S.; Sutti, A.; Greenhill, S.; Slezak, T.; Height, M.; Venkatesh, S., Rapid Bayesian optimisation for synthesis of short polymer fiber materials. *Scientific reports* **2017,** *7* (1), 1-10.
6. Abdelrahman, H.; Berkenkamp, F.; Poland, J.; Krause, A. In *Bayesian optimization for maximum power point tracking in photovoltaic power plants*, 2016 European Control Conference (ECC), IEEE: 2016; pp 2078-2083.
7. Kikuchi, S.; Oda, H.; Kiyohara, S.; Mizoguchi, T., Bayesian optimization for efficient determination of metal oxide grain boundary structures. *Physica B: Condensed Matter* **2018,** *532*, 24-28.
8. Khajah, M. M.; Roads, B. D.; Lindsey, R. V.; Liu, Y.-E.; Mozer, M. C. In *Designing engaging games using Bayesian optimization*, Proceedings of the 2016 CHI conference on human factors in computing systems, 2016; pp 5571-5582.
9. Lorenz, R.; Violante, I. R.; Monti, R. P.; Montana, G.; Hampshire, A.; Leech, R., Dissociating frontoparietal brain networks with neuroadaptive Bayesian optimization. *Nat. Commun.* **2018,** *9* (1), 1-14.
10. Frazier, P., A tutorial on Bayesian optimization. *arXiv: 1807.02811* **2018**.
11. Herbol, H. C.; Hu, W.; Frazier, P.; Clancy, P.; Poloczek, M., Efficient search of compositional space for hybrid organic–inorganic perovskites via Bayesian optimization. *npj Computational Materials* **2018,** *4* (1), 1-7.
12. Herbol, H. C.; Poloczek, M.; Clancy, P., Cost-effective materials discovery: Bayesian optimization across multiple information sources. *Materials Horizons* **2020**.
13. Yamashita, T.; Sato, N.; Kino, H.; Miyake, T.; Tsuda, K.; Oguchi, T., Crystal structure prediction accelerated by Bayesian optimization. *Physical Review Materials* **2018,** *2* (1), 013803.
14. Ju, S.; Shiga, T.; Feng, L.; Hou, Z.; Tsuda, K.; Shiomi, J., Designing nanostructures for phonon transport via Bayesian optimization. *Physical Review X* **2017,** *7* (2), 021024.
15. Ueno, T.; Rhone, T. D.; Hou, Z.; Mizoguchi, T.; Tsuda, K., COMBO: an efficient Bayesian optimization library for materials science. *Materials discovery* **2016,** *4*, 18-21.
16. Hashimoto, W.; Tsuji, Y.; Yoshizawa, K., Optimization of Work Function via Bayesian Machine Learning Combined with First-Principles Calculation. *The Journal of Physical Chemistry C* **2020,** *124* (18), 9958-9970.
17. Balachandran, P. V.; Kowalski, B.; Sehirlioglu, A.; Lookman, T., Experimental search for high-temperature ferroelectric perovskites guided by two-step machine learning. *Nat. Commun.* **2018,** *9* (1), 1-9.
18. Higgins, K.; Valleti, S. M.; Ziatdinov, M.; Kalinin, S. V.; Ahmadi, M., Chemical Robotics Enabled Exploration of Stability in Multicomponent Lead Halide Perovskites via Machine Learning. *ACS Energy Lett.* **2020,** *5* (11), 3426-3436.
19. Ling, J.; Hutchinson, M.; Antono, E.; Paradiso, S.; Meredig, B., High-dimensional materials and process optimization using data-driven experimental design with well-calibrated uncertainty estimates. *Integrating Materials and Manufacturing Innovation* **2017,** *6* (3), 207-217.

20. MacLeod, B. P.; Parlane, F. G.; Morrissey, T. D.; Häse, F.; Roch, L. M.; Dettelbach, K. E.; Moreira, R.; Yunker, L. P.; Rooney, M. B.; Deeth, J. R., Self-driving laboratory for accelerated discovery of thin-film materials. *Science Advances* **2020,** *6* (20), eaaz8867.
21. Min, K.; Cho, E., Accelerated discovery of potential ferroelectric perovskite via active learning. *Journal of Materials Chemistry C* **2020,** *8* (23), 7866-7872.
22. Rezaeianjouybari, B.; Sheikholeslami, M.; Shafee, A.; Babazadeh, H., A novel Bayesian optimization for flow condensation enhancement using nanorefrigerant: a combined analytical and experimental study. *Chem. Eng. Sci.* **2020,** *215*, 115465.
23. Park, S.; Na, J.; Kim, M.; Lee, J. M., Multi-objective Bayesian optimization of chemical reactor design using computational fluid dynamics. *Comput. Chem. Eng.* **2018,** *119*, 25-37.
24. Schweidtmann, A. M.; Clayton, A. D.; Holmes, N.; Bradford, E.; Bourne, R. A.; Lapkin, A. A., Machine learning meets continuous flow chemistry: Automated optimization towards the Pareto front of multiple objectives. *Chem. Eng. J.* **2018,** *352*, 277-282.
25. Burger, B.; Maffettone, P. M.; Gusev, V. V.; Aitchison, C. M.; Bai, Y.; Wang, X.; Li, X.; Alston, B. M.; Li, B.; Clowes, R., A mobile robotic chemist. *Nature* **2020,** *583* (7815), 237-241.
26. Granda, J. M.; Donina, L.; Dragone, V.; Long, D.-L.; Cronin, L., Controlling an organic synthesis robot with machine learning to search for new reactivity. *Nature* **2018,** *559* (7714), 377-381.
27. Guo, Z.; Wu, S.; Ohno, M.; Yoshida, R., Bayesian Algorithm for Retrosynthesis. *Journal of Chemical Information and Modeling* **2020,** *60* (10), 4474-4486.
28. Häse, F.; Roch, L. M.; Aspuru-Guzik, A., Chimera: enabling hierarchy based multi-objective optimization for self-driving laboratories. *Chemical science* **2018,** *9* (39), 7642-7655.
29. Häse, F.; Roch, L. M.; Aspuru-Guzik, A., Next-generation experimentation with self-driving laboratories. *Trends in Chemistry* **2019,** *1* (3), 282-291.
30. Kondo, M.; Wathsala, H.; Sako, M.; Hanatani, Y.; Ishikawa, K.; Hara, S.; Takaai, T.; Washio, T.; Takizawa, S.; Sasai, H., Exploration of flow reaction conditions using machine-learning for enantioselective organocatalyzed Rauhut–Currier and [3+ 2] annulation sequence. *Chem. Commun.* **2020,** *56* (8), 1259-1262.
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. Reker, D.; Hoyt, E. A.; Bernardes, G. J.; Rodrigues, T., Adaptive Optimization of Chemical Reactions with Minimal Experimental Information. *Cell Reports Physical Science* **2020,** *1* (11), 100247.
33. Kim, K.; Lee, W. H.; Na, J.; Hwang, Y.; Oh, H.-S.; Lee, U., Data-driven pilot optimization for electrochemical CO mass production. *Journal of Materials Chemistry A* **2020,** *8* (33), 16943-16950.
34. Wang, Y.; Xie, T.; France-Lanord, A.; Berkley, A.; Johnson, J. A.; Shao-Horn, Y.; Grossman, J. C., Toward Designing Highly Conductive Polymer Electrolytes by Machine Learning Assisted Coarse-Grained Molecular Dynamics. *Chem. Mater.* **2020,** *32* (10), 4144-4151.
35. Attia, P. M.; Grover, A.; Jin, N.; Severson, K. A.; Markov, T. M.; Liao, Y.-H.; Chen, M. H.; Cheong, B.; Perkins, N.; Yang, Z., Closed-loop optimization of fast-charging protocols for batteries with machine learning. *Nature* **2020,** *578* (7795), 397-402.
36. Doan, H. A.; Agarwal, G.; Qian, H.; Counihan, M. J.; Rodríguez-López, J.; Moore, J. S.; Assary, R. S., Quantum Chemistry-Informed Active Learning to Accelerate the Design and Discovery of Sustainable Energy Storage Materials. *Chem. Mater.* **2020**.
37. Dave, A.; Mitchell, J.; Kandasamy, K.; Wang, H.; Burke, S.; Paria, B.; Póczos, B.; Whitacre, J.; Viswanathan, V., Autonomous Discovery of Battery Electrolytes with Robotic Experimentation and Machine Learning. *Cell Reports Physical Science* **2020**, 100264.
38. Ebrahimi, M.; Ting, D. S. K.; Carriveau, R.; McGillis, A.; Young, D., Optimization of a cavern-based compressed air energy storage facility with an efficient adaptive genetic algorithm. *Energy Storage* **2020,** *2* (6), e205.
39. Poloczek, M.; Wang, J.; Frazier, P. In *Multi-information source optimization*, Advances in Neural Information Processing Systems, 2017; pp 4288-4298.

40. Swersky, K.; Snoek, J.; Adams, R. P. In *Multi-task bayesian optimization*, Advances in neural information processing systems, 2013; pp 2004-2012.
41. Lam, R.; Allaire, D. L.; Willcox, K. E. In *Multifidelity optimization using statistical surrogate modeling for non-hierarchical information sources*, 56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2015; p 0143.
42. Cardoso, T. N.; Silva, R. M.; Canuto, S.; Moro, M. M.; Gonçalves, M. A., Ranked batch-mode active learning. *Information Sciences* **2017,** *379*, 313-337.