By how much can closed loop frameworks accelerate computational materials discovery?

Lance Kavalsky,[1,*] Vinay I. Hegde,[2,*] Eric Muckley,[2] Matthew S. Johnson,[3] Bryce Meredig,[2,†] and Venkatasubramanian Viswanathan[1,‡]

[1]*Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213*

[2]*Citrine Informatics, 2629 Broadway, Redwood City, CA 94063*

[3]*Massachusetts Institute of Technology, Cambridge, MA 02139*

(Dated: September 30, 2022)

Abstract

Implementation of automation and machine learning surrogatization into closed-loop computational scientific workflows is an increasingly popular approach to accelerate materials discovery. However, the scale of the associated speedup from this paradigm shift over traditional manual approaches remains an open question. In this work we rigorously quantify the acceleration from each of the components within a closed-loop framework by probing four sources of speedup: (1) automation, (2) calculation runtime improvement, (3) guided design space search, and (4) machine-learning surrogatization. This is done through the timing of automated software and corresponding manual computational experiments. Stemming from a combination of the first three speedup sources, we estimate that acceleration of materials discovery by over 10× can be achieved. By introducing surrogatization into the loop, we estimate that this can be further improved to 15–20×. This work highlights the value in closed-loop approaches towards accelerating materials discovery.

Keywords: automated high-throughput DFT, sequential learning, computational materials discovery

I. INTRODUCTION

Discovery of materials is a central barrier to next-generation energy technologies such as more efficient and environmentally friendly electrochemical synthesis processes. One particular example is to substitute the environmentally harsh Haber-Bosch process used to synthesize ammonia by identification of candidate materials that can catalyze the reaction electrochemically [1, 2]. However, finding such optimal candidates efficiently remains a challenge due to the large size of the feasible candidate space [3]. Development of methods to accelerate this search, even within a relatively bounded design-space, is crucial to meet approaching climate goals.

These considerations have motivated significant research into new methods for accelerated materials discovery, both experimentally and computationally [4, 5]. In the context of experimental screening, much research focus has taken the form of robotic experimentation for applications such as searching for battery electrolytes [6], finding thermally stable perovskites [7], and optimizing battery charging protocols [8]. These studies tend to employ a

* These authors contributed equally to this work

† bryce@citrine.io

‡ venkvis@cmu.edu

combination of robots to automate experimental tasks and a learning agent to guide subsequent studies, closing the loop. However, the trade-off is that automated experimental setups are highly application specific and difficult to adapt to new applications where different tasks may be required. Thus experimental workflows show much promise, but are at present limited in terms of generalizability.

In contrast, fully computational workflows are appealing due to being mainly compute-limited and are relatively more flexible in terms of application. These workflows share some similarities to closed-loop experimental workflows, with computational calculations substituted for experiments and algorithms for iteratively selecting candidates from the design space. Adding new tasks to computational workflows demands only additional compute resources, rather than physical materials necessary to synthesize and test new candidates. This allows for improved modularity for transferring existing closed-loop software components between different materials workflows. The use of an iterative guided design space search has demonstrated encouraging results in terms of speeding up materials discovery [9–25]. Consequently, multiple computational closed-loop workflows have been developed for applications such as catalyzing electrochemical CO_2 reduction and hydrogen evolution [26], finding stable iridium oxide polymorphs [27], and discovering stable binary and ternary systems [28]. In each of these studies, the guided candidate searches showed notable improvement in finding promising candidate materials over a random search.

While computational closed-loop frameworks demonstrate a promising approach to accelerate materials discovery, quantification of their benefits over more traditional approaches remains challenging. In particular, the degree to which cumulative speedups of a fully autonomous closed-loop framework combine to accelerate materials discovery remains unclear. To our knowledge, a detailed breakdown of sources of acceleration along with relative quantitative impacts on speedup has not been previously explored.

In this study we quantify the acceleration estimates of a closed-loop computational framework for an electrocatalysis application. We probe two types of fully autonomous computational workflows (Figure 1): i) a closed-loop framework consisting of high-throughput density functional theory (DFT) calculations which feeds into a sequential learning (SL) algorithm that can select the next batch of candidate systems (thereby closing the loop) and ii) an extension of the previous framework where the cycle has produced enough DFT data that a machine learning (ML) surrogate can be trained to a desired accuracy and replace the heavy

DFT calculations. Four categories of acceleration are considered:

1. Comprehensive end-to-end automation of computational workflows
2. Runtime improvements of atomic compute tasks
3. Efficient search over vast design spaces using uncertainty-informed SL, and
4. Surrogatization of the most time-consuming tasks with ML models

Within each of these categories we estimate respective speedups and accumulate them into overall acceleration metrics. For end-to-end streamlining we estimate the attributed speedup through timing comparisons of the automated tasks and their manual analogues. In addition, we introduce a human-lag model to simulate user-related delays associated with manual job management on a computing cluster. Acceleration from improved runtimes of the compute-tasks are presented in terms of both calculator settings and initial structure guesses for DFT structural relaxations. This is done via calculations for relaxing OH onto the hollow sites of a sample single-atom alloy, $Ni_1/Cu(111)$. Efficiency improvements in design-space searching are presented here via a simulated SL search on a sample problem of optimizing binding energies of CO. Speedup from leveraging ML-surrogates for energy prediction is estimated via DFT training set size needed to reach the desired model accuracy. Finally, we then accumulate these results into an overall acceleration for workflows both excluding and including surrogatization. Through a combination of improvements in each of the above areas, we demonstrate a reduction in time to discover a new promising electrocatalytic material by 80-95% when compared to conventional approaches.

II. RESULTS

Each of the forms of acceleration described above can synergize to provide overall speedup in materials discovery. We benchmark the acceleration of each individual category through timing estimates of the relevant components both within a closed-loop automated workflow and for equivalent tasks when using the traditional approach. For quantifying the baseline materials discovery estimates, we use a combination of both modeling and manual timing experiments. For the automated tasks, we use the *AutoCat* (`https://github.com/aced-differentiate/auto_cat`) and *DFT-in-the-Cloud* (DFTitC) software packages

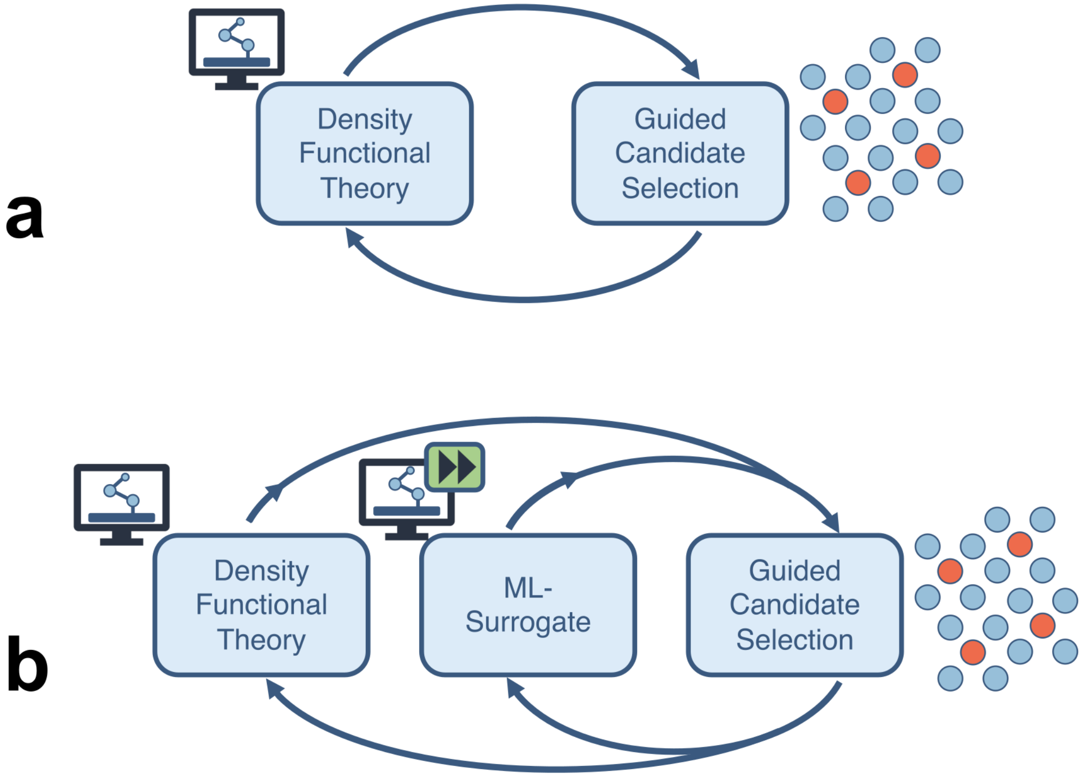

FIG. 1. Closed-loop materials discovery frameworks, a) without and b) with machine learning surrogates for the density functional theory calculations, to be considered in this work for acceleration quantification.

in tandem. Manual timing experiments use the *Atomic Simulation Environment* (ASE) [29] software package. Additional details are provided in Section V.

As an example design space, we chose the single-atom alloy (SAA) class of materials. SAAs are host transition-metals whose surface contains dispersed atoms of a different transition-metal species, and have shown much promise for electrocatalysis applications [30]. In particular, we focused our efforts on probing SAA systems that can catalyze electrochemical ammonia synthesis.

In the following subsections we discuss each of the individual acceleration categories and how their estimates were obtained. This is followed by acceleration estimates of the full workflow combining all sources of speedup to obtain a single acceleration estimate from the automated approach relative to the traditional baseline.

A. Automation of Computational Tasks and Workflows

Within a standard computational study, there are many time-consuming tasks related to preparing, managing, and analyzing DFT calculations. In Figure 2 we visualize a typical pipeline for a computational electrocatalysis study. Each of the boxes underneath a head symbol represents a task where user involvement is required in the traditional paradigm. This includes structure generation, DFT pre- and post-processing, and job management. Thus, every box in the pipeline that relies on user intervention is an opportunity for streamlining through automation.

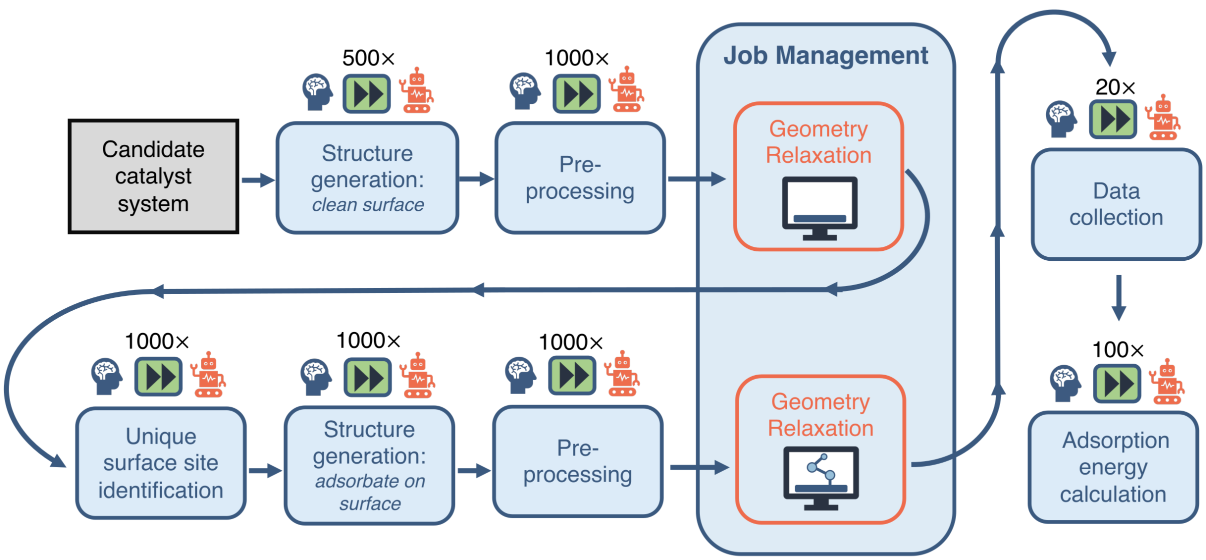

FIG. 2. Workflow for computational investigation of materials for electrocatalysis applications using density function theory. Blue boxes indicate computational tasks which typically require researcher input. Factors above each task indicate potential acceleration through automation. Orange boxes are geometry optimizations via density functional theory calculations.

To best benchmark the traditional workflow against an automated one, we define the same objective for both paradigms: calculation of the adsorption energies of OH on the SAA of a Ni atom embedded on a Cu 111 surface, designated as $Ni_1/Cu(111)$. This is further bounded to specifically include adsorption on all surface three-fold sites (6 in total). The goal is to mimic the scenario where an activity descriptor has already been identified for a specific electrochemical reaction, thereby collapsing performance predictions to the adsorption energy of a single adsorbate, as done previously [26]. As will be discussed later, this represents an optimization problem of a binding energy across a set of possible SAAs using SL to design the experiments. Previously, we have published methods to identify the most robust descriptors for a given reaction based on uncertainty quantification techniques [31, 32]. It should be noted that while automation generally replaces tasks that are on the order of seconds and minutes, the accelerations reported from this category free up the user to work on more analytical and constructive tasks, as elaborated in Section III.

All of the necessary steps to obtain the specified adsorption energies are highlighted in Figure 2. A comparison of the estimated time required for each task in the traditional approach and our automated approach is provided in Table I. Below, we outline the potential acceleration for each task via automation.

1. Candidate structure generation

As an input, DFT requires atomic scale structural representations of the candidate systems to be evaluated. Structure generation in the context of electrocatalysis consists of generation of the catalyst structure without any reaction intermediates, identification of all of the possible adsorbate sites, and placement of the reaction intermediates on the sites of interest. In this work we are not considering solvation effects. The first task corresponds to writing and executing scripts to generate the clean $Ni_1/Cu(111)$ slab via either ASE or *AutoCat* (corresponding to the manual and automated approaches, respectively), and comparing the relative timings. While ASE has functions tailored for the generation of some classes of systems, additional user involvement is necessary for those that are not currently implemented. As an example, ASE does not currently have functions geared specifically towards SAAs, and thus additional scripts are necessary to perform the doping of the pure slabs. To generate each SAA the dopant site needs to be identified, the substitution made, and spin polarization added to both the host and dopant as necessary. We can contrast this with automation software such as *AutoCat* which has a function built on top of ASE to streamline the generation of these SAA systems. Here, AutoCat can be viewed as fully automating ASE towards a specific application (in this case, SAAs). Moreover, the code is catered towards generating multiple SAAs through a single function call by the user that writes to disk in an organized, predictable fashion. By leveraging tools for streamlined candidate structure generation, a speedup of approximately 500× is observed. Thus, automation of this task, while relatively straightforward in some cases, does present an avenue for workflow acceleration.

Estimation of manual site identification for the second task of adsorbate placement requires measuring the time it takes a graduate student team member to identify all of the symmetrically unique surface sites of $Ni_1/Cu(111)$. This task becomes increasingly challenging from a user standpoint as the candidate catalyst becomes more elaborate, particularly with broken surface symmetries. For example, in the case of SAAs, the presence of the single-atom breaks many of the symmetries, and correctly identifying all unique sites by hand is nontrivial. Some sites that would be regarded as symmetrically equivalent on a uniform surface can no longer be regarded as such due to the substitution of the single-atom. In contrast, *AutoCat* identifies symmetry sites via the Delaunay Triangulation implementation within the `pymatgen` software package [33], providing a systematic automated approach to site identification that does not require user intervention. Comparison of the time required for a graduate student team member to identify all of the sites relative to the automated approach shows a speedup by a factor of 1000×. In summary, comparing the timings of these three tasks (catalyst surface generation, site identification, and adsorbate placement) highlights the effect of automation with regard to candidate structure generation tasks.

2. Density functional theory pre- and post-processing

For every catalyst structure generated, geometry optimizations via DFT calculations are required. The total energies from these relaxed structures can then be used to estimate properties of interest, such as adsorbate binding energy. Preparation for each of these calculations involves writing DFT input files and scripts to submit these calculations to high-performance computing (HPC) resources. The DFT input files contain all of the calculation settings to be used, such as the k-mesh and exchange-correlation functional. In addition, job submission scripts contain information about the requested computational resources on a cluster, including the number of cores needed and the wall-time before the job will be forcibly stopped. These scripts are necessary for every DFT calculation, and thus present an opportunity for automation. To obtain a baseline, we time a user performing both the above script writing tasks, i.e., generating DFT input files and batch submission scripts. This is then compared to the time required for the equivalent tasks within our DFTitC framework.

We observe that these automated tasks are approximately 1000× faster than their manual equivalent, again emphasizing that the automation of these DFT pre-processing tasks is a worthwhile effort.

Additionally, once the DFT calculations have successfully completed, the compilation of results and data can consume a significant amount of time. The user must read through each of the DFT output files, extract the desired information, and collect and organize this data. Scaling up to a large number of systems, and thus calculation outputs, this can quickly become a sizeable task. Here, we take timings of how long our DFTitC software takes to parse the output data and compare it to the time taken to manually read all of the output files and collect all of the data into a single spreadsheet. From automating this compilation procedure we observe a speedup by a factor of 20×.

Given reference states, adsorption energies can be calculated from the total energies. We thus compare the time required to calculate these adsorption energies within a spreadsheet to that of automated calculations, which we observe to be 100× faster when streamlined. This final post-processing step of calculating the adsorption energies is relatively quick regardless of the approach taken compared to the other steps considered in this workflow.

3. Workflow integration

In addition to the automation of structure creation and DFT pre- and post-processing as described above, the automation of the submission of batch jobs to HPC clusters, status monitoring, and general job management also provide opportunities for significant acceleration. DFT calculations of catalyst structures are computationally expensive and typically require active monitoring by the researcher. In particular, as these calculations can take variable lengths of time, they may demand user intervention. For example, this could be to fix errors or resubmit jobs, often at unpredictable times. This introduces "human lag" as it is not possible for the typical researcher to continuously monitor the status of all submitted DFT jobs. Here, human lag is modelled via a Monte Carlo sampling approach. First, days are subdivided into three different windows representing typical working hours, hours where some monitoring may occur, and hours where usually no monitoring would occur, with "checkpoints" in time defined for each. Next, a uniform distribution is assumed for the job finishing on any day of the week, without any preference for weekdays or weekends. Finally, we simulate the process of completion of a DFT job followed by research action at the nearest checkpoint in time, gathering statistics for a total of 10^6 DFT jobs, such as average lag per DFT job. In contrast, since job management within the fully-automated workflow is handled by a pipeline involving DFTitC and the `fireworks` [34] software package, there is no equivalent human lag, which enables significant acceleration.

Workflow step	Traditional	Automated	Acceleration
Catalyst structure generation			
Clean surface	16 min	2 s	$\sim$500x
Site identification	10 min	1 s	$\sim$1000x
Adsorbate placement	9 min	1 s	$\sim$1000x
DFT pre- and post-processing			
Generating DFT input and job management scripts	9 min	1 s	$\sim$1000x
Data collection	3 min	9 s	$\sim$20x
Adsorption energy calculation	2 min	1 s	$\sim$100x
DFT job submission and management			
Job resubmission and error handling	9 hr	–	–

TABLE I. Acceleration from automation of computational tasks and workflows.

B. Calculation Runtime Improvements

In the next category of acceleration, we quantify the speedup of calculation runtimes. Within our electrochemical materials discovery workflow, the primary physics-based simulation is DFT. As these calculations can be time-intensive, improving their runtimes is crucial in achieving significant acceleration.

In the case of adsorption structures, the initial guesses of the adsorbate geometry can play a key role. If the initial guess is far from the equilibrium geometry, more optimization steps will be required to achieve relaxation. Since each step requires a full DFT calculation to get the energy and forces, the initial guess should ideally be as close to the equilibrium as possible to decrease the overall calculation runtime. The total runtimes of geometry optimizations via DFT can also be heavily influenced by the choice of calculator settings, such as initial magnetic moment. Starting with a poor guess of the initial magnetic moment could result in longer time to achieve convergence in the DFT algorithm. To decouple the influences of both initial geometry guess and appropriate choice of calculator settings, we run four sets of relaxations for OH on all of the hollow sites of $Ni_1/Cu(111)$. We use two initial geometry guesses, one of which we call a chemically informed configuration as it is relatively close to the relaxed configuration. The initial height for this configuration is guessed based upon the covalent radii of the nearest neighbors of the anchoring O atom. For comparison, a chemically naive configuration is also considered where the initial geometry is further from the relaxed configuration, with the OH bond angle at 45° with the surface. Moreover, the initial height in this case is taken to be 1.5 Å above the surface. In addition to the different geometries, we also take two approaches to setting the initial magnetic moment of the single-atom. One approach is to choose the initial magnetic moment based on the ground-state magnetic moments of the single-atom species from ASE. This approach tailors the selection of initial magnetic moment to each particular system. Alternatively, we also test applying a default starting initial magnetic moment of 5.0 for the dopant, regardless of the identity of the single-atom species. This is a relatively naive approach as it does not incorporate details of the specific system for this choice of setting. In this specific case of $Ni_1/Cu111$, since the structure prefers to be in a spin-paired state (i.e., without spin-polarization), the former approach provides a guess closer to the actual spin-polarization for which the system converges. Our intention here is to highlight the impact of these aspects of the DFT calculations on acceleration and could stem from deterministic algorithms, an ML model, or another approach entirely.

In Figure 3 we visualize the accelerations of the DFT runtimes from both the calculation settings and initial geometries. Firstly, we observe relatively modest speedups from choice of calculator settings with average values of approximately 1.1× for both the naive and informed geometries. For these calculations, the system converges to the non-polarized state within the first few steps. Thus the observed speedup from choice of initial magnetic moment of the single-atom is mainly a reflection of these initial iterations while it finds the appropriate spin state, which often take the longest. On the other hand, we observe a larger acceleration effect from the initial geometry. When keeping the settings to be chosen naively or tailored while changing the initial geometry, we observe a speedup of 2.1× and 2.2×, respectively. The speedup described here can be mainly explained by the decrease in the number of steps required to reach the equilibrium configuration for a fixed optimization scheme. Using the tailored calculation settings, an average of approximately 33 and 16 geometry optimization steps with the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm are taken when starting from the chemically naive and informed geometries, respectively. Similarly, when using the naive calculation settings the average number of optimization steps for the chemically naive geometry is 34 and for the chemically informed geometry it is 17. Thus, methods to decrease the number of steps taken to reach equilibrium and shorten the DFT compute time at each step are highly desirable, and are an area of active development [35–40]. Combining both the improved initial geometry as well as the choice of calculator settings yields the largest runtime acceleration observed in this work of 2.3×, thus motivating the consideration of both variables within automated workflows. As will be discussed in greater detail in section II E, DFT takes up a substantial portion of the overall pipeline runtime, and thus this acceleration factor is indicative of potentially enormous improvements in overall acceleration.

Workflow step	Traditional	Automated	Acceleration
DFT calculation settings and initial structure guess			
Clean substrate relaxation	21 hr	18.5 hr	~1.1x
Substrate + adsorbate relaxation	46 hr	20 hr	~2.3x

TABLE II. Acceleration from calculation runtime improvements.

C. Efficient Design Space Search

Next, we estimate the acceleration resulting from use of a SL workflow for selecting and evaluating candidates in a design space of catalysts and compare it to that of traditional approaches. The SL workflow proceeds as follows: (1) start with an initial set of a small number of training examples of catalyst candidates and their properties; (2) build ML models using the initial set of training examples and predict the objective properties of all the candidates in the design space of interest; (3) use an acquisition function that considers

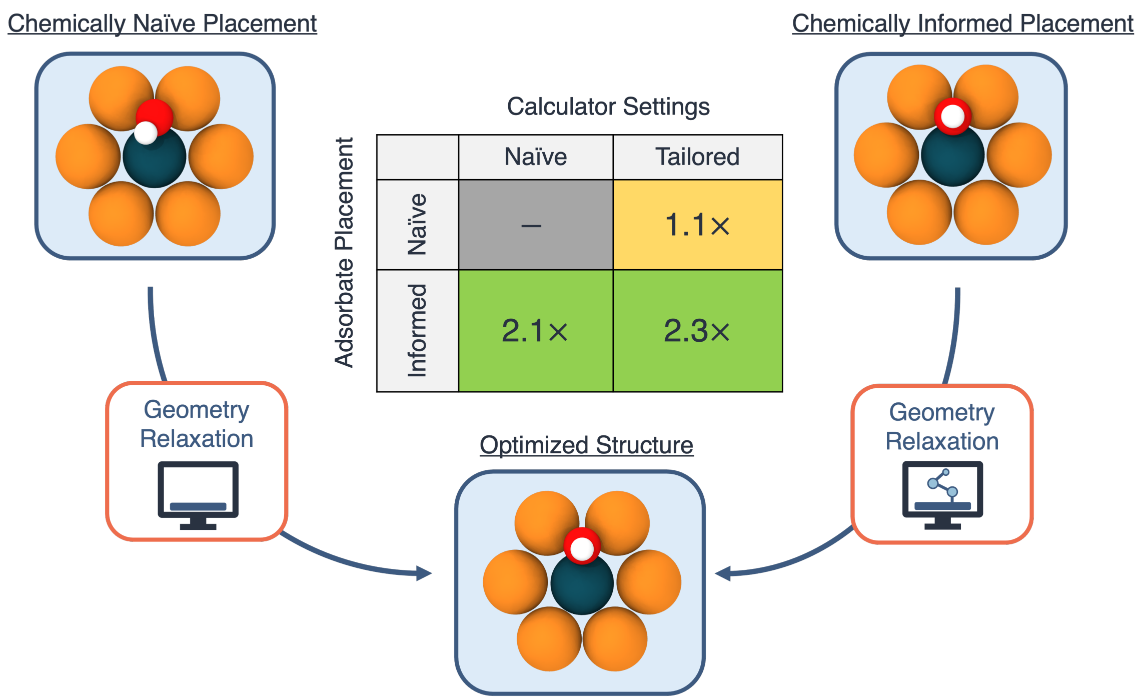

FIG. 3. Estimated density functional theory geometry optimization runtime accelerations. These are decoupled between chemically informed or naive initial geometries, and tailored or naive calculation settings. The largest factor of acceleration is observed when using an informed structure generation with tailored calculator settings.

model predictions and uncertainties to select the next candidate to evaluate; (4) evaluate the selected candidate and add it with its newly obtained label to the training set; (5) iterate steps 2–4 in a closed-loop manner until a candidate, or a certain number of candidates, with the target properties has been discovered. A detailed schematic of this workflow is presented in Figure 4. Such a strategy has been previously shown to be more efficient in sampling the design space to find novel candidates by a factor of $2-6\times$ over traditional grid-based searches or random selection of candidates from the design space [9–25].

For benchmarking the acceleration from SL for a typical catalyst discovery problem, we use a dataset of ∼300 bimetallic catalysts for CO_2 reduction [41]. The dataset contains ∼30 candidates with the target property of *CO adsorption energy on the catalyst surface inside a narrow window of [−0.7 eV, −0.5 eV]. We perform an SL simulation, starting with a small initial training set of 10 examples from the above dataset, and iterate in a closed-loop as described above until all the target candidates in the design space have been identified

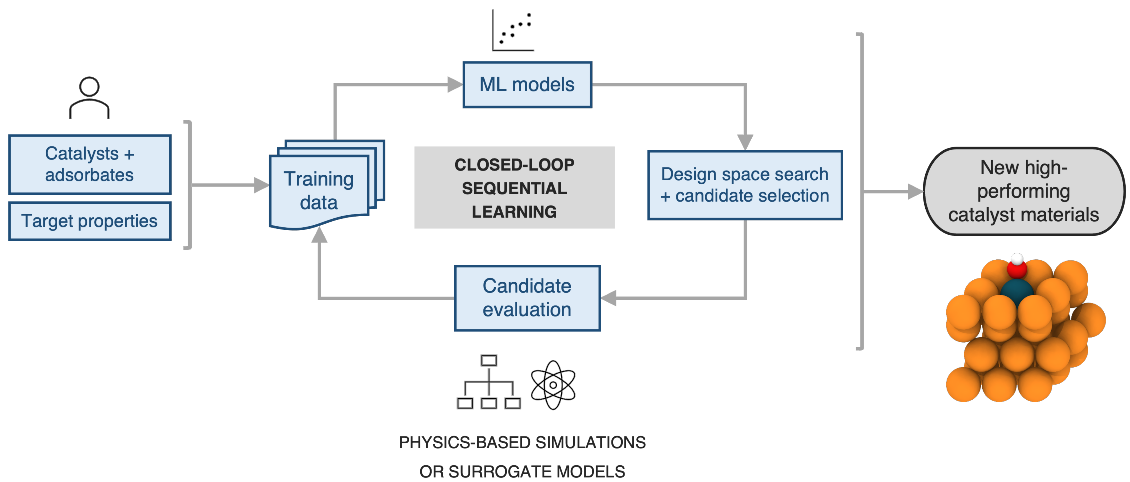

FIG. 4. A typical closed-loop sequential learning workflow for computational discovery of novel catalyst materials.

successfully, and benchmark the acceleration against random search. In particular, at each SL iteration, we build random forest-based models using the *lolo* software package [19], and predict the *CO adsorption energies of all candidates along with robust estimates of uncertainty in each prediction. The next candidate to evaluate is chosen based on the maximum likelihood of improvement (MLI) acquisition function. This function selects the system with the maximum likelihood of having an adsorption energy in the $[-0.7$ eV, -0.5 eV$]$ window, when considering both the predicted value as well as its uncertainty. Overall we find that such an SL-based workflow successfully identifies all $\sim$30 target candidates 3$\times$ faster than random search (Figure 5a). In addition, we note that the candidates surfaced by SL, on average, have properties closer to the target property window than those surfaced by random search, even when those candidates do not explicitly fall within the window (Figure 5b). In other words, in addition to discovering target candidates considerably more efficiently than random search, an SL-based approach surfaces potentially interesting candidates near the target window much more frequently than random search.

D. Surrogatization of Compute-Intensive Simulations

For the last category of acceleration, we estimate the extent of further possible speedup through the surrogatization of the most time-consuming tasks in the workflow. In particular,

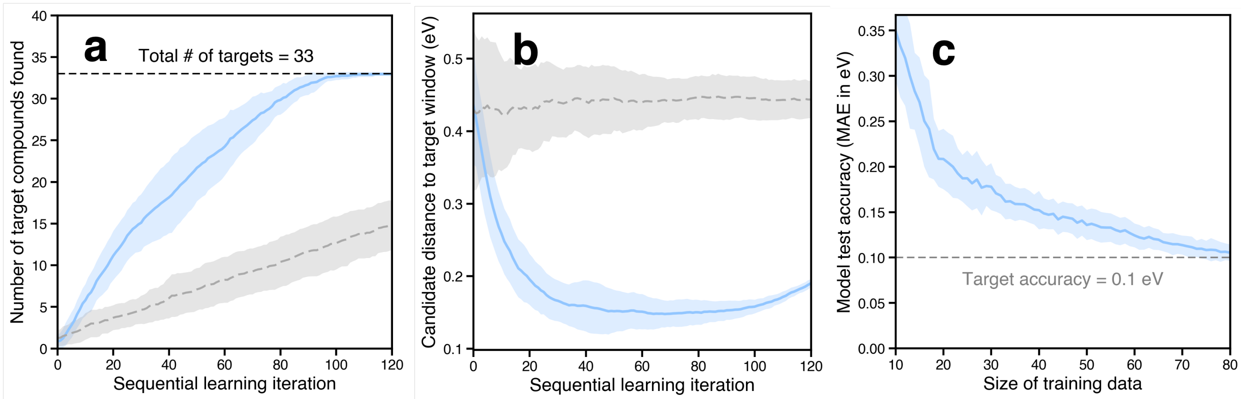

FIG. 5. Comparison of random search vs sequential learning (SL)-driven approach to find new bimetallic catalysts with a target property. (a) Overall, the SL-driven approach identifies all the 33 target candidates in the dataset within 100 iterations, $\sim 3\times$ faster than random search. (b) Candidates surfaced via SL lie much closer to the target window on average, when compared to those identified via random searching. (c) An SL-driven approach can help identify a much smaller number of examples that can be used to train ML surrogates to a desired accuracy, at a fraction of the overall dataset size. Here, the overall dataset has $\sim$300 candidates, and an ML model trained on only $\sim$25% of the candidates chosen via a SL-driven maximum uncertainty-based approach achieves the target accuracy.

the rate determining step of the closed-loop framework considered here is the calculation of the binding energies of adsorbates using DFT. ML models can be used as surrogates for physics-based simulations of material properties often at a fraction of the compute cost and with marginal loss in accuracy. The primary cost of building such ML surrogates for materials properties often lies in the generation of training data where such data does not exist, especially when the data generation involves compute-intensive physics-based simulations such as DFT. Here we estimate the size of such training data required to build and train ML surrogates with a target accuracy, especially when such training data is iteratively built using an SL-based strategy.

We use the dataset of bimetallic catalysts for CO_2 reduction mentioned in Section II C within a SL workflow to simulate an efficient, targeted training set generation scheme. Similar to the SL workflow employed in the search for novel catalyst materials in a design space of interest, we employ a closed-loop iterative approach to generate the training data and address model uncertainty. We start with a small initial training dataset of 10 systems, build random forest models to predict adsorption energies, and iteratively choose the next candidate to build the training data. With model accuracy in mind, we employ an acquisition strategy that optimizes for the most accurate ML model on average. In particular, at each iteration the candidate whose property prediction has the maximum uncertainty (MU) is selected to augment the training data. Inclusion of such a candidate results in the highest improvement of the overall accuracy of the ML model by targeting areas of the design space which are not well reproduced by the model. Using an accuracy threshold of interest, we then determine the fraction of the overall training data necessary for building useful ML surrogates. For instance, with a threshold of 0.1 eV (the typical difference between DFT and experimental formation energy values [42]), we estimate that accurate ML surrogates can be trained using a dataset generated via the above SL-strategy with ∼25% of the overall dataset size (Figure 5c). The accuracy metric here is calculated via a bootstrapping approach for the test set, with additional details provided in the Supplementary Information.

E. Overall Acceleration of the Full End-to-end Workflow

Finally, we aggregate the acceleration from the various steps in the workflow to estimate the overall speedup achieved. Here, we use the single-atom alloys (SAA) design space for calculating the overall estimates. We begin by estimating the size of such a design space. Limiting the design space to ∼30 transition metal hosts and dopants results in a total of $\sim {}^{30}C_2 \approx 900$ SAA systems. For each SAA system, typically a few (3–5) low-index surface terminations are considered. Moreover, the considered reaction intermediate can adsorb onto the catalyst surface at one of many possible symmetrically unique sites (up to 20–40 configurations), and all such possible intermediate configurations need to be considered in the design space. Overall, a typical SAA design space when fully enumerated can have up to 10^5–10^6 possibilities.

Using the above SAA design space, we apply the estimated time for each step in our overall end-to-end catalyst workflow as derived in the previous sections, using both traditional and automated methods (with and without surrogates), and calculate the overall speedup. From the automation of tasks and workflows, and runtime improvements alone, we achieve an acceleration of ∼10× (a reduction of ∼90%) over traditional materials design workflows. Further utilizing the ML surrogates (including the compute costs required to generate the training data) can result in an acceleration of up to $\sim$25$\times$ (a reduction of up to $\sim$96%) over traditional approaches.

Approach	Structure Generation	Substrate Calculation	Adsorbate Placement	Catalyst Calculation	Data Usage	Post-processing	Design Space Search Factor	Total Acceleration
Traditional	16 min	21 hr	18 min	72 hr/i.c.[1]	100%	5 min	1	
Automated	2 s	18.5 hr	2 s	20 hr/i.c.[1]	100%	10 s	0.33	10$\times$
+ Surrogates	2 s	–	2 s	20 hr/i.c.[1]	10-25%[2]	2 s	0.33	15-20$\times$

TABLE III. Overall acceleration benchmarks for the end-to-end workflows with and without surrogatization. We demonstrate a speedup of up to 10x with automation of tasks and runtime improvements, and a speedup of up to 25x upon using ML surrogates for the most compute-intensive DFT tasks. [1]i.c. = intermediate configuration (total # i.c. $\approx$ 200/catalyst system); "traditional" includes human lag estimates. [2]estimate from bimetallic catalyst dataset of the relative amount of DFT data needed to reach a target accuracy of 0.1 eV/adsorbate.

III. DISCUSSION

The results presented here have implications that reach beyond the reported factors of acceleration. Here, it is helpful to make a distinction between project time and researcher time. We consider project time as the time necessary to carry a project to completion. In other words, this is an accumulation of all the time spent towards achieving the tasks to reach the project goal. Thus, all the acceleration factors quantified above are with respect to this project time. Therefore, these workflows are anticipated to have a direct impact on project time to completion. In addition, by breaking down the acceleration benefits for each component of the workflows, estimates of project time acceleration for differing framework topologies (e.g. multi-scale evaluation) than those outlined here can be inferred.

On the other hand, researcher time can be interpreted as time spent from the frame of reference of the researcher on a given workday. The acceleration associated here is not directly quantified as we have done with project time. Instead, this acceleration is an indirect consequence of implementing these workflows. The most obvious example of this influence is through task automation. In the traditional paradigm, these tasks can become time-consuming, particularly as the scale of the project increases. Automation frees up valuable researcher time that would normally be occupied by the more mundane tasks. This allows the researcher to instead focus on more intellectually demanding tasks such as performing literature surveys and project formulation, improving research productivity.

Automating job management has the benefit of impacting both project time and researcher time. Since this form of automation facilitates running computational jobs around-the-clock, the human-lag as described by our model is entirely removed. If a job were to finish outside of working hours, there will not be any lag. This decreases the project time as described above. In the context of researcher time, this automation also has the added benefit of decreasing the need for regular job monitoring. Thus, during the day a researcher can devote more time towards other tasks.

A few additional observations regarding the nature of the baselines used to estimate the speed of traditional approaches in this work are warranted. First, for estimation of task timings such as input file generation for simulations and script generation to submit jobs on HPC resources, we use time estimates from a single researcher. The timings of such tasks are inherently variable, depending on the exact nature of the task, the researcher performing it, as well as the environmental setup in which it is performed. Similarly, natural delays associated with monitoring and managing ongoing computational jobs depend on the working habits of the researcher, the time-scale associated with each computation (e.g., those that take hours opposed to days or weeks to complete), and the availability or connectivity of the computational resources (e.g., on-site resources versus those that can be accessed remotely). Lastly, to estimate the acceleration from an intelligent exploration of the design space using sequential learning, we use random sampling as the benchmark. While random sampling is an excellent exploratory acquisition function [43] [include REF, e.g., several hyperparameter tuning algorithms use random search over a grid to find the optimal set within a fraction of full grid search: https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf. Bryce may have a reference handy that is more directly relevant for SL/design of experiments], it is not a substitute for traditional methods of design space exploration. Typically, traditional search approaches are influenced by prior knowledge, research directions within the community at the time, available resources, among other factors. We use random search here, not least because a model to predict a traditional materials design trajectory does not exist, to our knowledge, but also because it is widely-used as an unbiased exploratory baseline [9–13, 15–17, 19–25].

We want to emphasize that, given some of the variability in the baselines as discussed above, the goal of this work is to highlight the scale of acceleration that can be attributed to the several individual components in a computational materials design workflow. Moreover, we also aim to highlight the challenges associated with estimating such factors of acceleration, and not necessarily the raw factors themselves. Our work underscores the benefits of data collection and sharing, especially around time spent on research tasks, monitoring and managing medium- to high-throughput computational projects, implementing traditional approaches of materials discovery and design trajectories, and handling failed computations and experiments. We recommend a community-driven initiative towards such data collection and sharing efforts to bolster our understanding of the traditional baselines as well as to further contextualize the significant benefits of automation and ML-guided strategies.

IV. CONCLUSION

In this work we demonstrate that software automation and runtime improvements combined with a sequential learning-based closed-loop search over a design space for new catalysts can provide an overall acceleration of more than 10x (or more than 90% reduction in overall time/cost) over traditional approaches. Further, we estimate that such automation frameworks can have a significant impact on researcher productivity (20–1000×), direct compute costs (1.1–2.3×), and project/calendar time (>10–20×). This was estimated using 3 sources of acceleration. Through combination of manual computational experiments with timing of automated equivalent tasks, we provide speedup estimates stemming from each category. Automation of tasks can provide improved discovery time by streamlining tasks that usually need to be completed via user intervention. We also identify that significant speedup in terms of DFT runtime can be achieved through better initial prediction of the catalyst geometries and calculator settings. Moreover, the use of a sequential learning framework to guide design of experiments can accelerate discovery by a factor of 3, thereby dramatically decreasing the number of full loop iterations that need to be performed to reach a given design goal. We extend this analysis to include replacement of DFT calculations with machine-learning surrogates, another source of acceleration, and observe that this discovery speedup factor can be further improved to >15–20×. We believe that the results outline the immense benefits of introducing automation and machine learning into scientific discovery workflows, and motivate the increasingly widespread adoption of these methods.

V. METHODS

A. Workflow Topology

In this work we consider two different closed-loop topologies. The first is a two-stage process consisting of DFT calculations to calculate adsorption energies which are then fed into a SL model to iteratively guide candidate selection (Figure 1a). Within the DFT stage, multiple steps need to be taken for study of an electrocatalysis problem. Namely, a geometry relaxation of the clean candidate system, followed by a relaxation of all reaction intermediates onto the unique surface symmetry sites. These DFT calculations are taken to be automated within a pipeline. Here we use a combination of *AutoCat* (`https://github.com/aced-differentiate/auto_cat`) for automated structure generation and the *DFTitC* ecosystem for the calculations themselves. More details on these software packages are provided in Section V B. The SL stage then serves to guide the design space search by iteratively identifying candidates to evaluate on each loop. Additional details on the models used for this purpose are described in Section V D.

Another topology we consider is an extension of that described above with ML surrogates for the DFT calculations introduced into the loop (Figure 1b). In this scenario, the first few overall iterations proceed the same as before, except now as the DFT data is generated, a surrogate ML model is trained on the resulting data until a threshold test accuracy is reached. For these first number of iterations, candidates are selected with the intent of improving prediction accuracy. Once the threshold accuracy for the surrogate is met, all subsequent iterations of the loop will instead use the surrogate to obtain adsorption energies. From this point onward candidate selection is then focused on identifying promising materials using this trained ML model for continued exploration, as before in the loop without surrogatization. Motivation for such an approach stems from the computationally demanding nature of DFT calculations which is an inevitable bottleneck of these calculations. Swapping these heavy calculations for an essentially zero-cost alternative to obtain the same results is desirable in terms of overall speed. If desired, candidates identified via the ML model surrogate can be validated using DFT, but we are not including this aspect explicitly in our analysis. Thus, in this extended workflow we place emphasis on the number of DFT calculations necessary to obtain a surrogate meeting the minimum testing accuracy.

B. Automation Software

To create the atomic structures for the DFT calculations, we use *AutoCat*, a software package with tools for structure generation and sequential learning for electrocatalysis applications. This package is built on top of the Atomic Simulation Environment (ASE) [29] and `pymatgen` [33] to generate the atomic structures *en masse*, and write them to disk following an organized directory structure. To generate single-atom alloys, *AutoCat* has tailored functions for this purpose with optional selection of supercell dimensions, vacuum spacing, as well as number of bottom layers to be fixed, with appropriate defaults for each parameter. Moreover, through the use of `pymatgen`'s implementation of Delaunay triangulation [44], all of the unique symmetry sites on an arbitrary surface can be identified. Furthermore, initial heights of adsorbates are estimated through the covalent radii of the anchoring atom within a given adsorbate as well as its nearest neighbors. As the development of this package is part of an ongoing work, additional details will be reported in a future publication.

Once the catalyst with adsorbate systems have been generated by *AutoCat*, the crystal structures are used as input to an automated DFT pipeline that (a) generates input files for a DFT calculator (here we use GPAW[45, 46]), (b) executes DFT calculation workflows, and (c) parses successfully completed calculations and extracts useful information.

Automatic DFT input generation: We leverage the Python-based *dftinputgen* package (`https://github.com/CitrineInformatics/dft-input-gen`) to automate the generation of DFT input files from a specified catalyst/adsorbate crystal structure. In particular, we extend the *dftinputgen* package to support GPAW. For a given input crystal structure, the package provides sensible defaults to use for commonly-used DFT parameters based on prior domain knowledge for novice users as well as fine-grained control over each parameter for more experienced DFT practitioners. The package also implements, "recipes", sets of DFT parameters and values to be used as default depending on the properties of interest, e.g., ground-state geometry and electronic structure. The package outputs input files in a user-specified location that can be directly used by mature DFT packages as input for calculation.

Execution of DFT calculation workflows: We leverage the Python-based *fireworks* [34] package to both define complex sequences of DFT subcalculations necessary for electrocatalysis studies (e.g. clean surface relaxation followed by adsorption relaxation), and to create, submit, and monitor batch jobs on HPC resources that correspond to such a sequence of subcalculations. These scripts are part of an ongoing study and will be open-sourced.

Parsing output from DFT: After having completed DFT calculations on a large number of different candidate systems, key metrics such as total energy and forces would need to be extracted in bulk. To accomplish this task we have extended the previously-developed *pif-dft* (`https://github.com/CitrineInformatics/pif-dft`) and *dftparse* (`https://github.com/CitrineInformatics/dftparse`) packages to parse output generated via GPAW. Functions written for this package can look for a *.traj* file resulting from a successful GPAW run, in a specified directory. Once a traj file has been identified, it can be read using ASE to extract calculated properties of interest. This includes not only results such as total energy and forces, but also calculator settings such as the exchange-correlation implemented. These findings are then written into a Physical Information File (PIF) [47] (`https://citrine.io/pif`), a general-purpose materials data schema, for every calculation conducted.

C. First-Principles Calculations

All DFT calculations are conducted with the `GPAW` package [45, 46] via ASE [29]. The projector-augmented wave method is used for the interaction of the valence electrons with the ion cores. A target spacing of 0.16 Å is applied for the real-space grid, with a Monkhorst-Pack [48] k-mesh of $4 \times 4 \times 1$ for all surface calculations. For improved self-consistent field convergence, a smearing width of 0.05 eV is applied through the Fermi-Dirac distribution.

For the computational experiments adsorbing OH onto the $Ni_1/Cu(111)$ hollow sites described in Section II B, we employ two approaches for both the selection of calculator settings as well as initial geometry configuration. With regard to the calculator setting approaches, the tailored approach gives the dopant an initial magnetic moment based on the ground-state magnetic moment from ASE. On the other hand, the naive approach to calculator settings, where a uniform value is given regardless of species (in this case 5.0), is used for the dopant initial magnetic moment. In terms of the initial geometries, the chemically naive approach places OH at 45° with respect to the surface and an initial height of 1.5 Å above the surface. In contrast, the chemically informed initial structure has the OH bond as perpendicular to the surface plane. Moreover, the initial height for the chemically informed approach is an average based on the covalent radii of the nearest neighbor species at a given xy coordinate. All geometry optimizations are conducted via the BFGS algorithm as implemented in ASE.

D. Machine Learning Models

All ML models reported in this work are based on random forests [49], consistent with the previously-reported FUELS framework [19] and as implemented in the open-source *lolo* library [50]. Materials in the training dataset are transformed into the Magpie features [13], a set of descriptors generated using only the material composition, as implemented in the *matminer* package [51]. The uncertainty in a model prediction is determined using jackknife-after-bootstrap and infinitesimal jackknife variance estimators [52].

CONFLICTS OF INTEREST

There are no conflicts to declare.

ACKNOWLEDGEMENTS

The work presented here was funded in part by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0001211. L.K. acknowledges the support of the Natural Sciences and Engineering Research Council of Canada (NSERC). The authors thank Rachel Kurchin for helpful discussions around automation and acceleration estimation.

[1] B. H. Suryanto, H. L. Du, D. Wang, J. Chen, A. N. Simonov, and D. R. MacFarlane, Challenges and prospects in the catalysis of electroreduction of nitrogen to ammonia, Nature Catalysis **2**, 290 (2019).

[2] D. Chanda, R. Xing, T. Xu, Q. Liu, Y. Luo, S. Liue, R. A. Tufa, T. H. Dolla, T. Montini, and X. Sun, Electrochemical nitrogen reduction: recent progress and prospects, Chem. Commun. , (2021).

[3] Y. Kim, E. Kim, E. Antono, B. Meredig, and J. Ling, Machine-learned metrics for predicting the likelihood of success in materials discovery, npj Computational Materials **6**, 10.1038/s41524-020-00401-8 (2020), arXiv:1911.11201.

[4] D. P. Tabor, L. M. Roch, S. K. Saikin, C. Kreisbeck, D. Sheberla, J. H. Montoya, S. Dwaraknath, M. Aykol, C. Ortiz, H. Tribukait, C. Amador-Bedolla, C. J. Brabec, B. Maruyama, K. A. Persson, and A. Aspuru-Guzik, Accelerating the discovery of materials for clean energy in the era of smart automation, Nature Reviews Materials **3**, 5 (2018).

[5] R. Pollice, G. Dos Passos Gomes, M. Aldeghi, R. J. Hickman, M. Krenn, C. Lavigne, M. Lindner-D'Addario, A. Nigam, C. T. Ser, Z. Yao, and A. Aspuru-Guzik, Data-Driven Strategies for Accelerated Materials Design, Accounts of Chemical Research **54**, 849 (2021).

[6] A. Dave, J. Mitchell, K. Kandasamy, H. Wang, S. Burke, B. Paria, B. Póczos, J. Whitacre, and V. Viswanathan, Autonomous Discovery of Battery Electrolytes with Robotic Experimentation and Machine Learning, Cell Reports Physical Science **1**, 10.1016/j.xcrp.2020.100264 (2020), arXiv:2001.09938.

[7] Y. Zhao, J. Zhang, Z. Xu, S. Sun, S. Langner, N. T. P. Hartono, T. Heumueller, Y. Hou, J. Elia, N. Li, G. J. Matt, X. Du, W. Meng, A. Osvet, K. Zhang, T. Stubhan, Y. Feng, J. Hauch, E. H. Sargent, T. Buonassisi, and C. J. Brabec, Discovery of temperature-induced stability reversal in perovskites using high-throughput robotic learning, Nature Communications **12**, 1 (2021).

[8] P. M. Attia, A. Grover, N. Jin, K. A. Severson, T. M. Markov, Y. H. Liao, M. H. Chen, B. Cheong, N. Perkins, Z. Yang, P. K. Herring, M. Aykol, S. J. Harris, R. D. Braatz, S. Ermon, and W. C. Chueh, Closed-loop optimization of fast-charging protocols for batteries with machine learning, Nature **578**, 397 (2020).

[9] M. K. Warmuth, J. Liao, G. Rätsch, M. Mathieson, S. Putta, and C. Lemmen, Active learning with support vector machines in the drug discovery process, J. Chem. Inf. Comput. Sci. **43**, 667 (2003).

[10] A. Seko, T. Maekawa, K. Tsuda, and I. Tanaka, Machine learning with systematic density-functional theory calculations: Application to melting temperatures of single-and binary-component solids, Phys. Rev. B **89**, 054303 (2014).

[11] E. Pauwels, C. Lajaunie, and J.-P. Vert, A bayesian active learning strategy for sequential experimental design in systems biology, BMC Syst. Biol. **8**, 1 (2014).

[12] S. Chen, K.-R. G. Reyes, M. K. Gupta, M. C. McAlpine, and W. B. Powell, Optimal learning in experimental design using the knowledge gradient policy with application to characterizing nanoemulsion stability, SIAM-ASA J. Uncertain. **3**, 320 (2015).

[13] L. Ward, A. Agrawal, A. Choudhary, and C. Wolverton, A general-purpose machine learning framework for predicting properties of inorganic materials, npj Comput. Mater. **2**, 1 (2016).

[14] S. Kiyohara, H. Oda, K. Tsuda, and T. Mizoguchi, Acceleration of stable interface structure searching using a kriging approach, Jpn. J. Appl. Phys. **55**, 045502 (2016).

[15] E. V. Podryabinkin and A. V. Shapeev, Active learning of linearly parametrized interatomic potentials, Comput. Mater. Sci. **140**, 171 (2017).

[16] A. M. Gopakumar, P. V. Balachandran, D. Xue, J. E. Gubernatis, and T. Lookman, Multi-objective optimization for materials discovery via adaptive design, Sci. Rep. **8**, 1 (2018).

[17] R. Yuan, Z. Liu, P. V. Balachandran, D. Xue, Y. Zhou, X. Ding, J. Sun, D. Xue, and T. Lookman, Accelerated discovery of large electrostrains in batio3-based piezoelectrics using active learning, Adv. Mater. **30**, 1702884 (2018).

[18] R. E. Brandt, R. C. Kurchin, V. Steinmann, D. Kitchaev, C. Roat, S. Levcenco, G. Ceder, T. Unold, and T. Buonassisi, Rapid photovoltaic device characterization through bayesian parameter estimation, Joule **1**, 843 (2017).

[19] J. Ling, M. Hutchinson, E. Antono, S. Paradiso, and B. Meredig, High-dimensional materials and process optimization using data-driven experimental design with well-calibrated uncertainty estimates, Integr. Mater. Manuf. Innov. **6**, 207 (2017).

[20] H. C. Herbol, W. Hu, P. Frazier, P. Clancy, and M. Poloczek, Efficient search of compositional space for hybrid organic–inorganic perovskites via bayesian optimization, npj Comput. Mater. **4**, 1 (2018).

[21] A. D. Sendek, E. D. Cubuk, E. R. Antoniuk, G. Cheon, Y. Cui, and E. J. Reed, Machine learning-assisted discovery of solid li-ion conducting materials, Chem. Mater. **31**, 342 (2018).

[22] B. Rohr, H. S. Stein, D. Guevarra, Y. Wang, J. A. Haber, M. Aykol, S. K. Suram, and J. M. Gregoire, Benchmarking the acceleration of materials discovery by sequential learning, Chem. Sci. **11**, 2696 (2020).

[23] Z. Del Rosario, M. Rupp, Y. Kim, E. Antono, and J. Ling, Assessing the frontier: Active learning, model accuracy, and multi-objective candidate discovery and optimization, J. Chem. Phys. **153**, 024112 (2020).

[24] A. G. Kusne, H. Yu, C. Wu, H. Zhang, J. Hattrick-Simpers, B. DeCost, S. Sarker, C. Oses, C. Toher, S. Curtarolo, *et al.*, On-the-fly closed-loop materials discovery via bayesian active learning, Nat. Comm. **11**, 1 (2020).

[25] A. E. Gongora, B. Xu, W. Perry, C. Okoye, P. Riley, K. G. Reyes, E. F. Morgan, and K. A. Brown, A bayesian experimental autonomous researcher for mechanical design, Sci. Adv. **6**, eaaz1708 (2020).

[26] K. Tran and Z. W. Ulissi, Active learning across intermetallics to guide discovery of electrocatalysts for CO2 reduction and H2 evolution, Nature Catalysis **1**, 696 (2018).

[27] R. A. Flores, C. Paolucci, K. T. Winther, A. Jain, J. A. G. Torres, M. Aykol, J. Montoya, J. K. Nørskov, M. Bajdich, and T. Bligaard, Active Learning Accelerated Discovery of Stable Iridium Oxide Polymorphs for the Oxygen Evolution Reaction, Chemistry of Materials **32**, 5854 (2020).

[28] J. H. Montoya, K. T. Winther, R. A. Flores, T. Bligaard, J. S. Hummelshøj, and M. Aykol, Autonomous intelligent agents for accelerated materials discovery, Chemical Science **11**, 8517 (2020).

[29] A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. B. Jensen, J. Kermode, J. R. Kitchin, E. L. Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. B. Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, and K. W. Jacobsen, The atomic simulation environment—a python library for working with atoms, Journal of Physics: Condensed Matter **29**, 273002 (2017).

[30] R. T. Hannagan, G. Giannakakis, M. Flytzani-Stephanopoulos, and E. C. H. Sykes, Single-Atom Alloy Catalysis, Chemical Reviews **120**, 12044 (2020).

[31] L. Kavalsky and V. Viswanathan, Robust Active Site Design of Single-Atom Catalysts for Electrochemical Ammonia Synthesis, The Journal of Physical Chemistry C **124**, 23164 (2020).

[32] D. Krishnamurthy, V. Sumaria, and V. Viswanathan, Maximal Predictability Approach for Identifying the Right Descriptors for Electrocatalytic Reactions, Journal of Physical Chemistry Letters **9**, 588 (2018), arXiv:1709.02875.

[33] S. P. Ong, W. D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V. L. Chevrier, K. A. Persson, and G. Ceder, Python materials genomics (pymatgen): A robust, open-source python library for materials analysis, Computational Materials Science **68**, 314 (2013).

[34] A. Jain, S. P. Ong, W. Chen, B. Medasani, X. Qu, M. Kocher, M. Brafman, G. Petretto, G.-M. Rignanese, G. Hautier, D. Gunter, and K. A. Persson, FireWorks: a dynamic workflow system designed for high-throughput applications, Concurrency and Computation: Practice and Experience **27**, 5037 (2015).

[35] J. Yoon and Z. W. Ulissi, Differentiable optimization for the prediction of ground state structures (DOGSS), Physical Review Letters **17**, 173001 (2020).

[36] J. R. Boes, O. Mamun, K. Winther, and T. Bligaard, Graph Theory Approach to High-Throughput Surface Adsorption Structure Generation, Journal of Physical Chemistry A **123**, 2281 (2019).

[37] E. Garijo Del Río, S. Kaappa, J. A. Garrido Torres, T. Bligaard, and K. W. Jacobsen, Machine learning with bond information for local structure optimizations in surface science, The Journal of chemical physics **153**, 234116 (2020), arXiv:2010.09497.

[38] S. Deshpande, T. Maxson, and J. Greeley, Graph theory approach to determine configurations of multidentate and high coverage adsorbates for heterogeneous catalysis, npj Computational Materials **6**, 1 (2020).

[39] J. Musielewicz, X. Wang, T. Tian, and Z. Ulissi, Finetuna: Fine-tuning accelerated molecular simulations, arXiv preprint arXiv:2205.01223 (2022).

[40] E. G. del Río, J. J. Mortensen, and K. W. Jacobsen, Local bayesian optimizer for atomic structures, Physical Review B **100**, 104103 (2019).

[41] X. Ma, Z. Li, L. E. Achenie, and H. Xin, Machine-learning-augmented chemisorption model for co2 electroreduction catalyst screening, The journal of physical chemistry letters **6**, 3528 (2015).

[42] S. Kirklin, J. E. Saal, B. Meredig, A. Thompson, J. W. Doak, M. Aykol, S. Rühl, and C. Wolverton, The open quantum materials database (oqmd): assessing the accuracy of dft formation energies, npj Computational Materials **1**, 1 (2015).

[43] J. Bergstra and Y. Bengio, Random search for hyper-parameter optimization, Journal of Machine Learning Research (2012).

[44] J. H. Montoya and K. A. Persson, A high-throughput framework for determining adsorption energies on solid surfaces, npj Computational Materials **3**, 1 (2017).

[45] J. J. Mortensen, L. B. Hansen, and K. W. Jacobsen, Real-space grid implementation of the projector augmented wave method, Phys. Rev. B **71**, 035109 (2005).

[46] J. Enkovaara, C. Rostgaard, J. J. Mortensen, J. Chen, M. Dułak, L. Ferrighi, J. Gavnholt, C. Glinsvad, V. Haikola, H. A. Hansen, H. H. Kristoffersen, M. Kuisma, A. H. Larsen, L. Lehtovaara, M. Ljungberg, O. Lopez-Acevedo, P. G. Moses, J. Ojanen, T. Olsen, V. Petzold, N. A. Romero, J. Stausholm-Møller, M. Strange, G. A. Tritsaris, M. Vanin, M. Walter, B. Hammer, H. Häkkinen, G. K. H. Madsen, R. M. Nieminen, J. K. Nørskov, M. Puska, T. T. Rantala, J. Schiøtz, K. S. Thygesen, and K. W. Jacobsen, Electronic structure calculations with GPAW: a real-space implementation of the projector augmented-wave method, Journal of Physics: Condensed Matter **22**, 253202 (2010).

[47] K. Michel and B. Meredig, Beyond bulk single crystals: a data format for all materials structure–property–processing relationships, MRS Bull. **41**, 617 (2016).

[48] H. J. Monkhorst and J. D. Pack, Special points for Brillouin-zone integrations, Physical Review B **13**, 5188 (1976).

[49] L. Breiman, Random forests, Mach. Learn. **45**, 5 (2001).

[50] C. Informatics, Lolo, `https://github.com/CitrineInformatics/lolo` (2017).

[51] L. Ward, A. Dunn, A. Faghaninia, N. E. Zimmermann, S. Bajaj, Q. Wang, J. Montoya, J. Chen, K. Bystrom, M. Dylla, *et al.*, Matminer: An open source toolkit for materials data mining, Comput. Mater. Sci. **152**, 60 (2018).

[52] S. Wager, T. Hastie, and B. Efron, Confidence intervals for random forests: The jackknife and the infinitesimal jackknife, J. Mach. Learn. Res. **15**, 1625 (2014).