

A modular active learning framework for autonomous atomic layer deposition

Maier Alghalayini

Lawrence Berkeley National Laboratory <https://orcid.org/0000-0002-4071-0272>

Tim Kodalle

Lawrence Berkeley National Laboratory <https://orcid.org/0000-0002-8792-9669>

Edward S. Barnard

esbarnard@lbl.gov

Lawrence Berkeley National Laboratory

Research Article

Keywords:

Posted Date: September 16th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11039646/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

A modular active learning framework for autonomous atomic layer deposition

Maher B. Alghalayini^{1,2,†}, Tim Kodalle^{1,†}, Arian Gashi¹, Aleksandr Razumtcev¹, Mythili Surendran¹, Shaul Aloni¹, Adam M. Schwartzberg¹, and Edward S. Barnard^{1,*}

¹*Molecular Foundry, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA, USA*

²*Applied Mathematics and Computational Research Division, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA, USA*

[†]*These authors contributed equally to this work.*

^{*}*Corresponding author: esbarnard@lbl.gov*

Abstract

Optimizing gas-phase thin-film deposition processes remains a significant experimental challenge due to the high dimensionality of the synthesis parameter space and the complex, nonlinear interactions among process parameters. Here, using plasma-enhanced atomic layer deposition (PE-ALD) as a model system, we present a closed-loop materials acceleration platform that combines a fully autonomous synthesis tool, in situ characterization, and a Gaussian process-based active learning framework to autonomously and efficiently map synthesis-property relationships across high-dimensional parameter spaces. The active learning framework employs an information-theory-based acquisition function to sequentially identify the most informative experiments, systematically reducing model uncertainty with each iteration. We apply this framework to the deposition of TiO₂ using ALDBot, a custom-built automated PE-ALD tool with full programmatic control and equipped with in situ spectroscopic ellipsometry for real-time growth rate monitoring. In a 4-dimensional synthesis parameter sub-space, the active learning framework clearly outperforms random sampling in predictive accuracy and information gain, converging on a model that reflects the underlying synthesis-property relationship and accurately predicts the growth rate across the entire parameter space with significantly fewer experiments. Expanding to the full 12-dimensional parameter space requires a larger number of experiments for complete parameter space mapping; however, as little as 200 experiments enable surprisingly robust predictions, which are validated against independent experimental sweeps. More importantly, the trained Gaussian process model further enables physically interpretable outputs, including process window identification and parameter importance rankings, enabling further targeted process optimization. These results demonstrate that closed-loop integration of automated experimentation, in situ characterization, and active learning constitutes a powerful strategy for autonomous exploration of gas-phase deposition processes, broadly applicable beyond the specific material system and deposition technique studied here.

1 Introduction

Conventional process optimization for material fabrication typically consists of an iterative sequence of literature- and experience-informed experiment planning, execution, and characterization. Because these processes are slow, labor-intensive, and heavily dependent on prior knowledge, the materials community has started shifting towards autonomous, high-throughput synthesis strategies. Significant efforts have leveraged natural language processing and text mining of the scientific literature, combined with high-throughput computation, to identify and contextualize new material candidates.[1–3] In addition, robotic deposition and characterization tools increase experimental throughput, while machine learning (ML) and artificial intelligence (AI) frameworks improve experiment planning, execution, and analysis.[4–7] The most mature realization of this paradigm is the materials acceleration platform (MAP), combining ML frameworks with automated fabrication and (in-situ) characterization to autonomously select, execute, and analyze experiments with minimal human intervention.[8–10] To date, however, these combined efforts have largely focused on solution-based and combinatorial methods. ML and experimental automation individually have been broadly applied to gas-phase deposition processes, such as atomic layer deposition (ALD);[11–15] however, fully autonomous closed-loop gas-phase deposition MAPs are scarce,[16, 17] and to the best of our knowledge no ALD-MAPs have been reported.

ALD is a widely used thin-film deposition technique that can produce highly conformal, uniform films with sub-monolayer-thickness control, making it indispensable in semiconductors, microelectronics, energy storage, catalysis, and biomedical coatings.[18–20] Deposition proceeds via self-limiting “ALD cycles”: chemisorption of a metal-organic precursor followed by conversion to the target material (typically a metal oxide or nitride) in an activation step. The process window achieving this self-limiting conformal growth is referred to as the ALD window: the range of synthesis conditions over which the growth per cycle remains constant. Outside this window, the self-limiting behavior breaks down, and the growth rate deviates from its characteristic constant value.[21, 22] Among ALD variants, plasma-enhanced ALD (PE-ALD) is particularly flexible, enabling high-quality deposition at lower substrate temperatures through plasma rather than thermal activation.[23, 24] However, PE-ALD performance is highly sensitive to synthesis parameters, with complex, non-linear interactions among them.[25, 26] Identifying the ALD window and optimal process conditions through conventional one-parameter-at-a-time approaches can therefore be very time-consuming, making it an ideal candidate for a MAP-driven approach to mapping synthesis-property relationships efficiently.

Realizing such a MAP requires a model capable of learning these synthesis-property relationships from limited experimental data. Data-driven methods, such as neural network ensembles, random forests, and Gaussian processes, offer a promising route in high-dimensional parameter spaces, each exhibiting different tradeoffs between scalability, interpretability, and data efficiency.[27–29] Neural network-based approaches, for instance, scale well to large, high-dimensional datasets but typically require substantial training data and provide less reliable uncertainty estimates. In the data-scarce regime characteristic of materials synthesis, Gaussian processes combined with active learning have gained particular traction for their ability to produce accurate predictions from small datasets

and, crucially, to provide calibrated uncertainty estimates that distinguish between experimental variability and uncertainty arising from data scarcity.[12, 13, 30–32] This uncertainty quantification is essential for guiding experiment selection: an information-theory-based acquisition function directs sampling toward regions of the synthesis parameter space with the highest epistemic uncertainty, enabling systematic and efficient exploration.[33–35]

In the present work, we combine this active learning strategy with ALDBot, an in-house-built automated PE-ALD tool with full Python-based programmatic control and in situ spectroscopic ellipsometry for continuous growth-rate measurement, presenting a closed-loop PE-ALD-MAP. Using an experiment-campaign approach, we test the autonomous feedback loop framework across two separate campaigns, using subsets of the processing parameter space for PE-ALD TiO₂ thin films. In a first campaign, we aim to accurately map synthesis-property relationships across the entire synthesis parameter space (exploration) rather than identify an optimal point within it (exploitation). We show that the framework outperforms random-sampling baselines, maps the ALD window, and identifies optimal deposition conditions. In a second campaign, we test how the active learning strategy can provide valuable insights into parameter spaces that are too large for simple exploration within the given experimental budget. Here, we show how ranking parameter importance and providing physical insight into the deposition mechanism can give insights into the underlying synthesis-property relationships and reduce the complexity of future process optimization – even without complete mapping of the parameter space.

2 Results and Discussion

2.1 Autonomous deposition

To enable the autonomous exploration of PE-ALD processes, we developed a deposition system in which every relevant process parameter can be set and varied programmatically: ALDBot exposes a comprehensive set of parameters, spanning all steps of the plasma-enhanced ALD cycle, and controls them through a fully automated workflow capable of executing experimental campaigns without human intervention. A schematic and some details of the hardware of the custom-built, fully autonomous ALDBot used in this work are provided in Figure S1 in the supporting information (SI). The process steps performed during each automated deposition, together with the parameters controlling each step, are summarized in Figure 1a. Typically, an ALD growth cycle comprises six sequential steps, each guided by up to three parameters. In the first step, the vacuum chamber is conditioned for the ALD cycle and some general parameters are set to their default values for the first half cycle (Steps 1 through 3): For the duration of *precursor preparation time*, p_1 , the chamber is pumped to and held at the *process pressure*, p_2 , regulated by a closed-loop throttle valve between chamber and turbo pump and two separate argon flows into the chamber. The two flows are determined by *argon process flow rate*, p_3 , the Ar-flow into the chamber via the plasma source, and *precursor purge flow rate*, p_5 , the argon flow via the precursor line into the chamber. In the second step, the ALD precursor Tetrakis(dimethylamido)titanium (TDMAT) is introduced by dosing it from

the precursor source into the ALD chamber. The amount of precursor available is determined by the duration of this step (*precursor dose time*, p_4) as well as the temperature of the precursor source (fixed in this study). Note that the Ar-flow p_5 is turned off 10 ms before the dosing step, during the dosing step, and for an additional waiting time after the precursor valve is closed again (*precursor purge time*, p_6), to prevent TDMAT backflow during dosing and to evacuate the line afterwards. Third, the remaining precursor in the line is flushed into and out of the chamber using the *precursor purge flow rate*, p_5 for the same amount of time as the previous wait time, the *precursor purge time*, p_6 . Fourth, the chamber is flushed with oxygen in order to prepare the plasma reaction (parameters include *plasma preparation time*, p_7 , *oxygen flow rate*, p_8 , and *plasma pressure*, p_9 , i.e., the chamber pressure during the second half cycle). Note that the *argon process flow rate*, p_3 and *precursor purge flow rate*, p_5 are set to zero for Steps 4 and 5. Fifth, the plasma is ignited and used to react with the precursor previously chemisorbed on the substrate (parameters are *plasma duration*, p_{10} , and the *power set point*, p_{11} of the radio frequency (RF) source). Sixth, another purging step removes any remaining oxygen and gaseous reaction products for the duration specified by *plasma purge time*, p_{12} at *plasma pressure*, p_9 and with both argon flows restored. In total, 12 parameters are varied. Substrate and precursor temperatures were held fixed in this study to avoid time-consuming temperature equilibration. In addition, we used a constant window purge with argon to minimize deposition on the windows used for spectroscopic ellipsometry.

In addition, we decided to execute up to 200 experiments per single substrate to be able to fully utilize the setup’s automatic capabilities. This approach does not necessarily reflect standard fabrication conditions, where one would use a fresh silicon wafer for each deposition; however, we chose it to enable high-throughput experimentation in the absence of an efficient automated sample-exchange mechanism. To balance the trade-off between fresh growth conditions for each experiment and high-throughput deposition, we performed 25 ALD cycles per experiment (i.e., per set of parameters) to obtain a robust estimate of the growth rate independent of prior deposition. Furthermore, the data from the first 9 cycles of each experiment were discarded to exclude inter-experiment effects.[36]

2.2 A closed-loop active learning workflow

To elevate ALDBot from an automated deposition tool to an autonomous MAP, we integrated it into an autonomous feedback loop that designs, executes, and analyzes experiments without human intervention. This automatic feedback approach not only minimizes the time required to perform the experiments but also identifies the “most useful” experiments (see next paragraph for more details) before carrying them out, and analyzes the resulting output data. As shown in Figure 1b, the loop consists of four sequential steps: autonomously executing experiments using ALDBot, characterizing the resulting thin film via in situ ellipsometry and extracting the growth rate, fitting a Gaussian process model to the updated dataset, and selecting the next experiment by optimizing the corresponding acquisition function over the parameter space. The autonomous feedback loop is initialized by randomly selecting 5 distinct combinations of synthesis parameters. ALDBot then

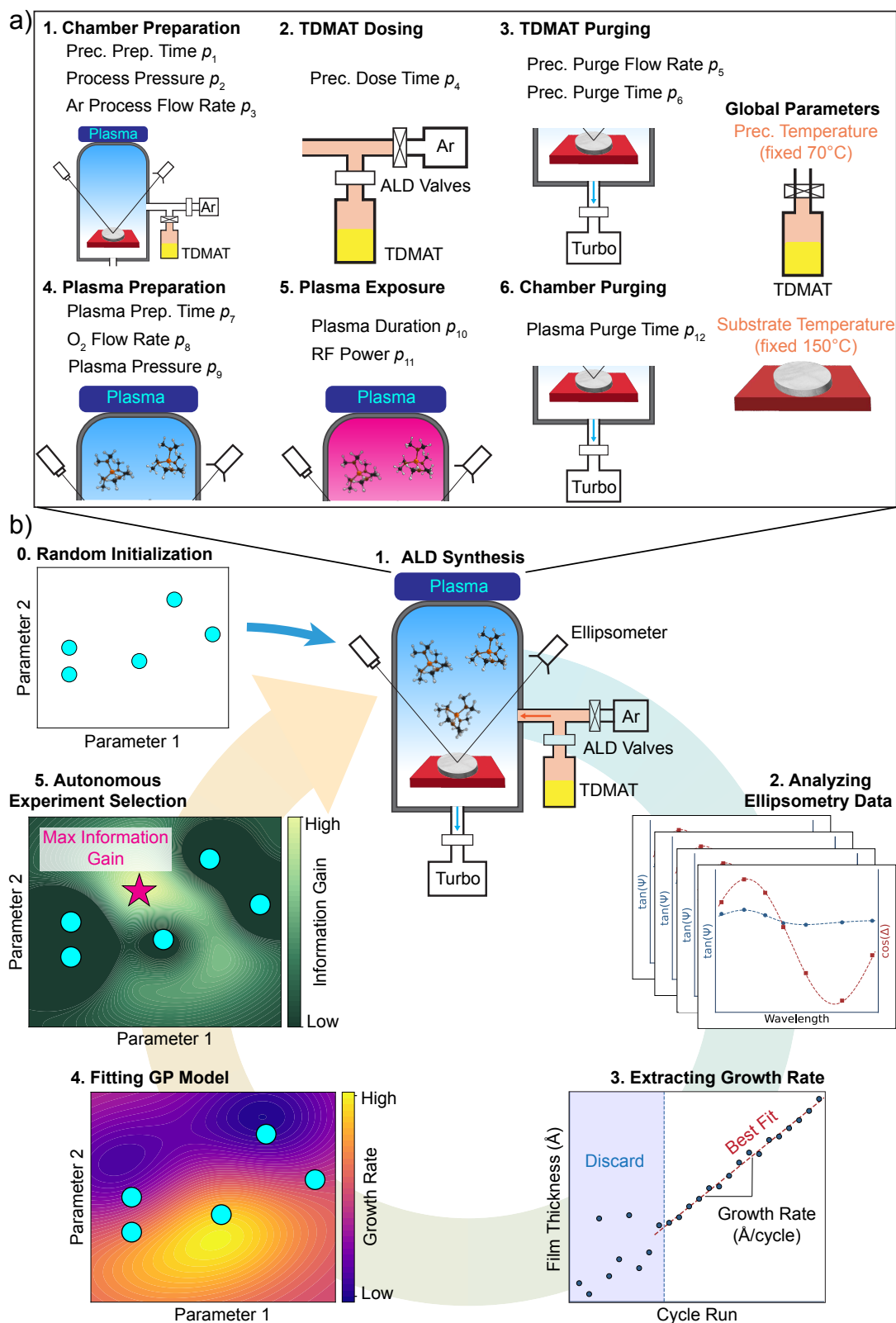


Figure 1. a) Schematic overview of a PE-ALD cycle using ALDBot and the process parameters in each of the six steps making up a complete ALD cycle and dictating the growth rate. Process parameters written in orange were not varied in this study due to their slow equilibration time. b) Machine learning-guided closed-loop workflow for autonomous exploration of the ALD parameter space. The loop begins with a set of randomly initialized experiments (1), which are deposited by ALDBot with in situ ellipsometry (2). The ellipsometry data are analyzed (3) to extract a growth rate for each experiment (4). A Gaussian process model is fit to the accumulated data (5), and an acquisition function selects the next experiment by maximizing the expected information gain (6); the selected conditions are fed back to ALDBot, closing the loop. Cyan points denote individual experiments in the parameter space.

autonomously receives these parameter sets and performs the corresponding experiments, executing 25 deposition cycles per parameter set. During deposition, the layer thickness of the material is measured in real time using in situ ellipsometry. The spectral ellipsometry data are fit to extract the layer thickness as a function of cycle number, and the growth rate for each experiment is obtained as the slope of this thickness-versus-cycle relationship. The growth rates from all experiments performed so far are then used to fit a Gaussian process model with the gpCAM Python package [37], which characterizes the current state of knowledge about the relationship between the synthesis parameters and the monitored property (growth rate). Using the fitted model, the total correlation acquisition function is then optimized throughout the synthesis parameter space to identify the next most informative experiment to perform. The newly chosen point is then fed back to ALDBot to execute the next experiment, restarting the loop. The process is repeated for a fixed budget of N experiments. Note that due to this modular approach, each of the loop’s components, namely the deposition process, the in situ characterization method, and the ML model, can be easily varied.

In this work, we implemented the above-described active learning strategy using a Gaussian process model and an information-theoretic acquisition function. The chosen acquisition function selects experiments expected to yield the largest reduction in the model’s predictive uncertainty across the parameter space by quantifying how much each candidate experiment is expected to improve the current state of knowledge. Specifically, we use the total correlation acquisition function, which identifies candidate points whose predicted outcomes are least correlated with both previously observed data and with each other, thereby maximizing information gain for global exploration. Note that we use a constant prior mean for the Gaussian process model; i.e., the model has no prior expectations regarding the growth-rate trend in the synthesis parameter space (see Methods for details).

Considering the large number of synthesis parameters available to ALDBot, exploring the synthesis parameter space using naive, brute-force approaches is experimentally infeasible. A grid sampling approach with, for instance, 10 points per parameter would result in $10^{12} = 1$ trillion experiments, and even expertise-driven design-of-experiment approaches would require several hundred data points. And while the latter approach is experimentally feasible (and has been proven successful in the ALD community), ML approaches like the one described above can further reduce the number of experiments needed to provide a comprehensive understanding of the space. In addition, unsupervised, autonomous approaches can significantly reduce the amount of time needed for such an experimental campaign. However, even with sophisticated ML approaches, scaling to high-dimensional spaces can still be costly.[38–40] The number of experiments needed to sufficiently cover an experimental space using Bayesian optimization, specifically, still scales exponentially.[38] While exploitation of a 12-dimensional parameter space as the one described in Figure 1 might still be feasible within the given experimental budget of 200 experiments per campaign, comprehensive exploration, which is the goal of this study, would require a much higher number of data points. Therefore, we will first benchmark our active learning strategy when fully exploring a four-dimensional subspace, and then analyze its performance across the entire 12-dimensional space. Note that throughout this work, we

distinguish between the active learning strategy and the autonomous feedback loop as two distinct components of the MAP. The active learning strategy refers specifically to the modeling component, i.e., the Gaussian process model and the acquisition function that selects which experiment to perform next. The autonomous feedback loop encompasses the automated ALDBot system that executes, characterizes, and provides feedback on each experiment without human intervention.

2.3 Case study: 4D synthesis parameter space

To evaluate the performance of the active learning strategy for a subset of 4 synthesis parameters, we fixed all but four parameters for the deposition of ALD-TiO₂ using a TDMAT precursor onto a silicon substrate. The selected parameters creating the 4-dimensional (4D) synthesis parameter sub-space composed of p_4 , *precursor dose time* with [minimum, maximum] limits of [100, 6000] ms, p_6 , *precursor purge time* ([10, 6000] ms), p_7 , *plasma preparation time* ([10, 4000] ms), and p_{12} , *plasma purge time* ([10, 6000] ms). The set values of the remaining parameters are summarized in Table S1 in the SI. The active learning campaign consisted of 100 experiments: 5 random initialization points plus 95 experiments selected sequentially by the acquisition function, one per iteration. As a baseline, we ran a random sampling campaign of the same size—the same 5 initialization points plus 95 randomly selected experiments—so that both campaigns comprise 100 experiments. In both campaigns, a Gaussian process model is fit to the accumulating data in exactly the same way; the campaigns differ only in how each new experiment is selected—by optimizing the acquisition function (active learning) or by random choice.

Figure 2a illustrates the evolution of the Gaussian process model predictions throughout the experimental campaign. From left to right, the plots show the predicted growth rate as a function of p_4 , *precursor dose time*, i.e., a one-dimensional slice through the 4D parameter space, after 10 (left), 20 (middle), and 100 (right) experiments. We use the term ‘slice’ to refer to a lower-dimensional cut through the high-dimensional space providing a view of the model function across a single parameter (here, p_4 , *precursor dose time*, while the remaining three parameters are fixed at $p_6 = 6$ s, $p_7 = 4$ s, and $p_{12} = 4$ s). Experimental variation of that parameter across the slice (in the next section) will be referred to as a ‘parameter sweep’. The red line in Figure 2a represents the predicted growth rate, and the shaded region represents the 95% confidence interval computed from the Gaussian process model’s predicted uncertainty. Notably, even after just 10 runs (of which 5 were random), the Gaussian process model accurately captures the shape of the growth rate as a function of p_4 , *precursor dose time*. Adding 10 runs significantly reduces the uncertainty of the model and shifts the magnitude of the predicted function closer to its value at the end of the experiments, while the additional 80 runs further reduce the uncertainty and adjust small features in the model, e.g., the dip of the prediction after 20 runs at high dose times. After 100 experiments we can clearly identify the ALD window for this given slice, the processing window for efficient atomic layer growth (see highlighted region in the rightmost panel of Figure 2a). Remarkably, no experiments with these exact parameters were performed during the active learning or included in the model’s training data, yet the model has interpolated this ALD window behavior from its data.

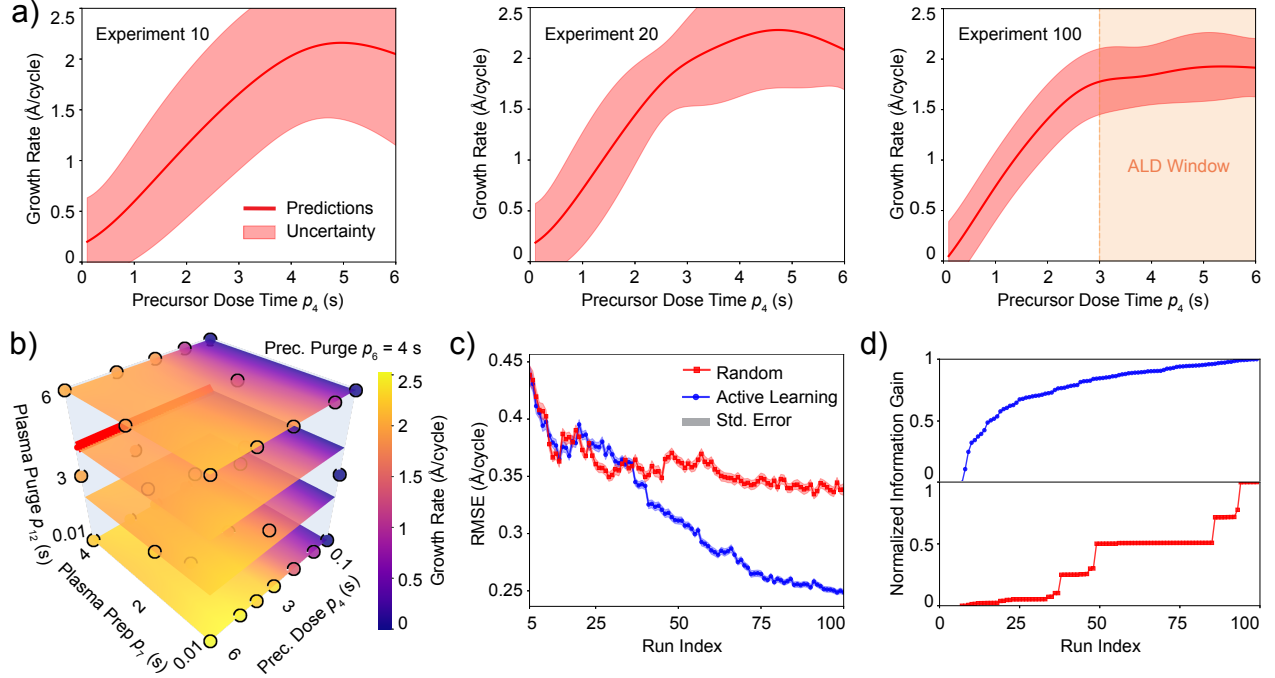


Figure 2. a) Evolution of the Gaussian process model by active learning and the corresponding uncertainty for a 1D-slice through the 4D-subspace after 10, 20, and 100 experiments, obtained by varying p_4 , precursor dose time, while holding p_6 , precursor purge time, p_7 , plasma preparation time, and p_{12} , plasma purge time fixed at 6 s, 4 s, and 4 s, respectively. b) 3D-slice through the same subspace after 100 experiments, obtained by fixing p_6 , precursor purge time at 4 s, comparing the Gaussian process model resulting from the active learning campaign to the actual data points taken in this subspace. The red line marks the location of the 1D slice shown in panel a, i.e., the line along which p_4 is varied with p_7 and p_{12} held fixed at 4 s each, as in panel a. c), d) RMSE and normalized information gain evolution over 100 experiments for active learning (blue) and random (red) sampling approaches.

Figure 2b shows a three-dimensional slice through the parameter space, fixing p_6 , precursor purge time at 4 s, with the color map representing the growth rate as predicted by the Gaussian process model after completing the 100 experiments (the colored data points indicate the experimentally determined growth rate of the experiments performed in this 3D sub-space). The bold red line marks the location within this 3D slice corresponding to the 1D slice shown in Figure 2a, i.e., the line along which p_4 , precursor dose time is varied while p_7 , plasma preparation time and p_{12} , plasma purge time are kept fixed at 4 s each. The sparse distribution of data points across the cube highlights the data efficiency of the active learning approach. From the 3D slice, we further see that both p_7 , plasma preparation time and p_{12} , plasma purge time show a negligible effect on the growth rate within the probed process condition space, as evidenced by the nearly uniform color across these dimensions. For a complete view of the synthesis-property relationship, Figure S2 in the supplementary material shows the full 4D space across all values of p_6 , precursor purge time, along with the corresponding uncertainty maps. The predictions confirm that p_6 , precursor purge time has a minor but non-negligible effect on the growth rate compared to p_4 , precursor dose time. Additionally, the uniformly low uncertainty across the space demonstrates that the active learning campaign has successfully characterized the synthesis-property relationship within this experimental

synthesis subspace. We also note that the exploration approach tends to explore the edges and corners of the parameter space, as observed in this 3D slice.

Figures 2c and d show the evolution of the root mean square error (RMSE) and normalized cumulative information gain as a function of the number of experiments for the active learning (blue) and random (red) sampling campaigns. RMSE measures the predictive accuracy of the model when tested on unseen data, while information gain quantifies the magnitude of change of the Gaussian process model upon adding each new experiment, computed as the Kullback–Leibler (KL) divergence between the posterior predictive distributions before and after the new observation. Because the two campaigns explore parameter space in different orders, the raw magnitude of this per-step divergence is not directly comparable between them; the cumulative curves in Figure 2d are therefore each normalized to their own final value, so that the comparison of interest is the rate at which each method approaches its asymptote rather than the total accumulated change (see Supplementary Figure S3 for the unnormalized comparison).

The active learning campaign outperforms random sampling across both metrics. Both approaches start with comparable RMSE values until approximately 35–40 experiments, after which the active learning approach separates clearly from random sampling, achieving a steadily decreasing RMSE that reaches approximately 0.25 Å/cycle by the end of the campaign, while the random sampling RMSE fluctuates and plateaus at a substantially higher value of approximately 0.35 Å/cycle. This inconsistency is also evident in the step-like behavior of the random-sampling information gain curve in Figure 2d, where long, flat regions indicate redundant experiments that contribute little new information, punctuated by sudden jumps when a randomly chosen point lands in a sparsely sampled region and forces a large corrective update to an otherwise overconfident posterior. The active learning approach, in contrast, shows a rapid rise in normalized information gain over the first iterations, reflecting the information-theory-based acquisition function’s ability to consistently target high-uncertainty regions of the parameter space before such miscalibration can accumulate. The curve then plateaus early, as the model converges and the system becomes well characterized, with each subsequent experiment yielding diminishing returns. This indicates that active learning reaches a well-calibrated model in substantially fewer experiments than random sampling.

2.4 Generalization to 12D synthesis parameter space

Having established that the active learning strategy can efficiently and comprehensively characterize a given synthesis parameter space with a feasible dimensionality, we next apply the framework to the full 12-dimensional synthesis parameter space shown in Figure 1a. Accounting for this substantial increase in dimensionality, we fully exhausted our experimental budget of 200 experiments per campaign and increased the number of random initialization runs to 15. The experimental limits of each synthesis parameter are summarized in Table S2 in the SI.

As discussed above, a significantly larger number of experiments would be needed to comprehensively map a 12-dimensional parameter space; however, we aim to ask whether a high-uncertainty Gaussian process model trained on a sparsely sampled space retains practically valuable information

for an experimentalist. Benchmarking the RMSE evolution of the active learning strategy against random sampling confirms that both approaches show largely similar predictive power, with only an insignificant improvement for active learning (see Figure S4). As shown in the previous section, further experimentation is expected to significantly quicker reduce the RMSE using the active learning strategy compared with random sampling. However, even after these 200 experiments, the Gaussian process model captures substantially more information than its RMSE alone would suggest. In particular, having trained a model across the full 12D parameter space, we can extract insights into underlying synthesis-property relationships that would be difficult to identify otherwise. To do this in a rigorous way, we use SHapley Additive exPlanations (SHAP, see methods for more details).[41, 42] Intuitively, SHAP asks, for each individual experiment: how much did each synthesis parameter push the model’s growth rate prediction up or down relative to the model’s average prediction across all experiments? It answers this by systematically testing how the prediction changes as parameters are added to the model input, one at a time and in every possible order, and averaging each parameter’s contribution across all such orderings. This yields a single number (the SHAP value) for every parameter in every experiment, which can be both summed to give the total prediction and used to rank parameters by their overall influence.

Figure 3a summarizes this analysis for all 12 synthesis parameters. Each row corresponds to a parameter, ordered from most influential (top) to least influential (bottom) by the average magnitude of its SHAP values across all training points. Each dot represents one experiment, with its horizontal position (SHAP value, in Å/cycle, the same units as the predicted growth rate) indicating how much that parameter increased or decreased the predicted growth rate for that experiment, and its color indicating whether the parameter was set to a high (red) or low (blue) value in that experiment, using the same min-max normalized representation used for model training (see Methods). This allows us to quantitatively rank parameters by their influence on the model’s predictions and also to see how each parameter drives the growth rate up or down. Note that this ranking reflects what the trained model has learned from the sampled region of parameter space, rather than a proven, ground-truth statement about parameter importance for the underlying physical process; we would expect a model trained on any sufficiently informative dataset, including randomly sampled data, to yield a qualitatively similar ranking. While the resulting influences on the growth rate may seem somewhat obvious for parameters like the p_4 , *precursor dose time*, it is less obvious for others (e.g., p_{10} , *plasma duration*), enabling further, targeted experiments.

Interestingly, only one of the four parameters we selected for the 4D case, p_4 , ranks among the top four in the objective, data-driven SHAP ranking; p_6 , p_7 , and p_{12} (also highlighted in bold in Figure 3a) all rank in the bottom half of the parameters probed. This is consistent with the 4D Gaussian process model itself (Figure 2b and S4), which also showed a comparatively minor sensitivity to p_6 and a near-negligible dependence on p_7 and p_{12} , indicating that the 4D active learning campaign had also revealed this same hierarchy of parameter importance.

To further experimentally test the predictive accuracy of the sparse-data-based Gaussian process model and to test the result of the SHAP analysis, we conducted parameter sweeps over p_2 , p_4 ,

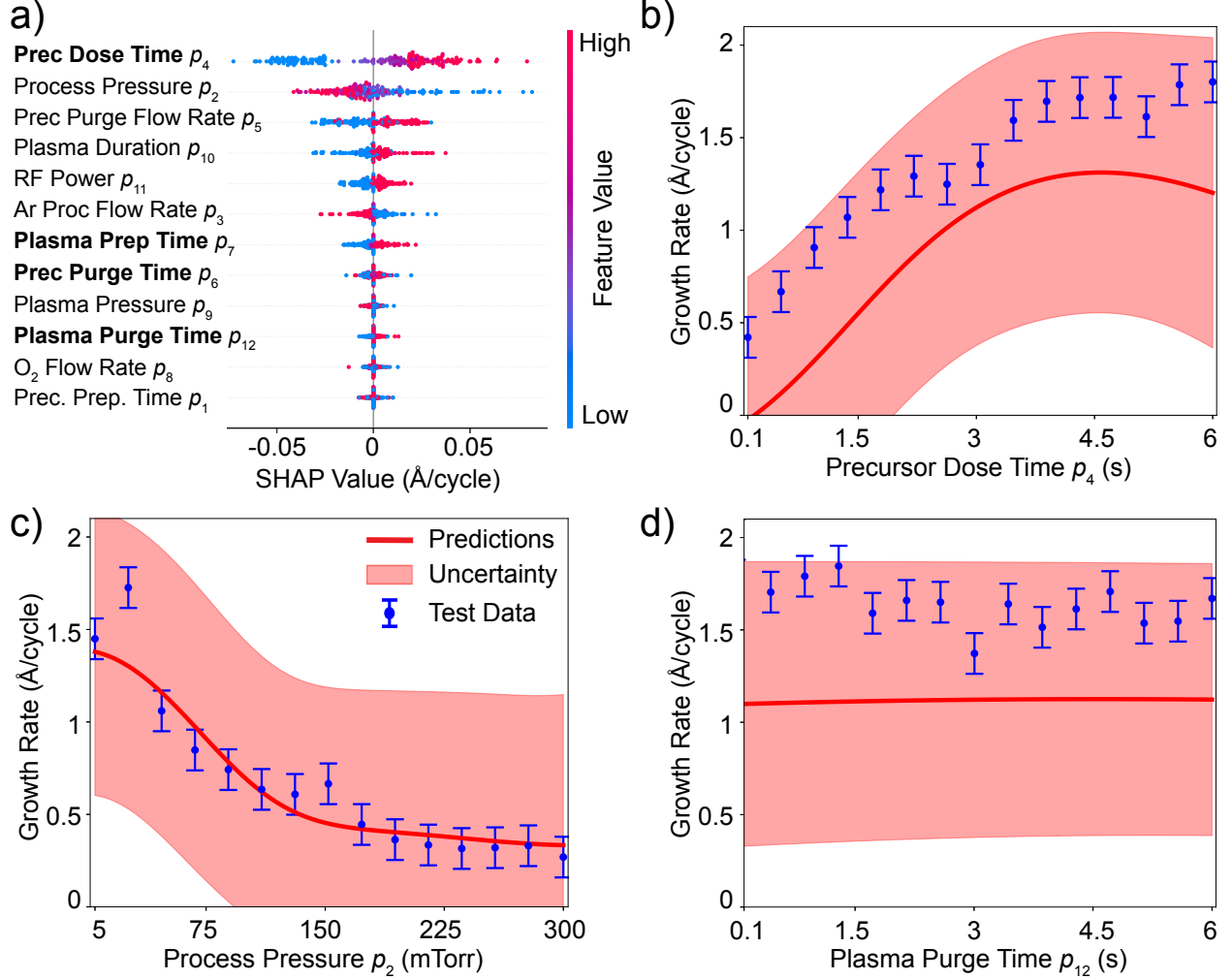


Figure 3. Prediction performance and feature importance extracted from the Gaussian process model for the 12D case. (a) SHAP (SHapley Additive exPlanations) summary plot illustrating the relative importance of each process parameter on the Gaussian process model output trained on data from the active learning approach. Each point represents an experiment in the training set, with color indicating the value of the feature (normalized parameter p_i) and horizontal position indicating the magnitude and direction of its impact on the predicted growth rate. Bold labels indicate the four parameters explored in the 4D case (Figure 2). (b–d) Gaussian process model predictions validated against independent test sweeps conducted on the two most influential process parameters (b) p_4 , precursor dose time and (c) p_2 , process pressure, and one parameter that is not influential (d) p_{12} , plasma purge time, as identified by the SHAP analysis. The solid red line and shaded region represent the Gaussian process predicted mean and 95% confidence interval, respectively, while the blue points with error bars denote experimental measurements from dedicated validation sweeps.

and p_{12} — the two most influential synthesis parameters and one low-ranking parameter — while keeping all other parameters fixed. Here, a sweep refers to a traditional optimization experiment in which all but one parameter are held fixed and that one parameter is varied across its allowed range to identify its direct impact on the output variable. Importantly, the three sweep experiments were conducted separately and were not included in the Gaussian process model’s training dataset, thereby serving as independent validation points. Figure 3b–d overlays the experimental sweep data (markers) with the Gaussian process model predictions (line and shaded region). For all three parameters, the experimental sweeps remain within the model’s 95% confidence interval throughout the whole probed range. This highlights that, for both parameters that vary comparatively strongly and those that vary only slightly, the model can qualitatively (and even quantitatively, within the uncertainty) predict the thin-film growth rate from a very sparse dataset. Furthermore, the strong dependence of the growth rate on p_2 and p_4 , together with the flat response to p_{12} , qualitatively confirms the SHAP analysis results. We therefore conclude that the active learning framework yielded an informative model of the growth rate landscape, enabling predictions even at an early stage with high model uncertainty. While these high-uncertainty predictions alone may not be able to inform optimal experimental conditions, they can be used to inform more targeted optimization either by sparse sweeps or re-running an active learning strategy on a subspace.

Finally, to test whether our model correctly predicts the ALD window as a region of stable and efficient ALD growth of TiO_2 , we select a representative but time-efficient parameter combination at the edge of the predicted ALD window, deposit around 5 nm of TiO_2 using this parameter set, and validate the chemical composition of the resulting thin film. We used the information provided by the model across the whole space to select a parameter combination that lies in the ALD window while prioritizing low values (i.e., short times) for all temporal parameters, highlighting how knowledge of the whole space enables smart parameter choices for individual process optimization. The XPS spectra of the Ti 2p and O 1s regions (Figure 4) confirm an atomic ratio of 33.6% titanium and 66.4% oxygen. The small shoulder at higher binding energies in the O 1s spectrum corresponds to the native oxide on the Si substrate. These results confirm the deposition of phase-pure TiO_2 via ALDBot, further validating the results presented above as an efficient, fast, and reliable approach to explore a synthesis parameter space.

Conclusions

The complexity of modern materials synthesis and especially gas-phase deposition processes, combined with the vast and complex parameter spaces that govern them, presents a fundamental challenge to conventional, intuition-driven process development, especially in novel complex material systems. Here, we demonstrate that this challenge can be effectively addressed by combining automated experimentation, in situ characterization, and active learning into a closed-loop process. By integrating a Gaussian process-based active learning framework with an information-theory-based acquisition function into a fully automated experimental platform, we demonstrated how synthesis-

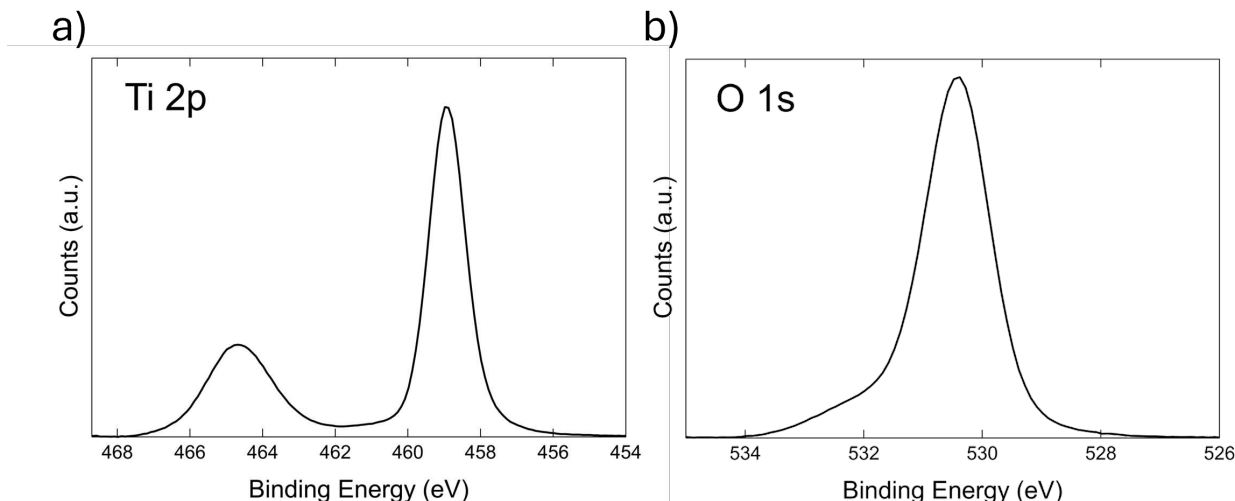


Figure 4. a) and b) show XPS spectra of the Ti 2p and O 1s peaks, respectively, taken on a roughly 5 nm thick TiO_2 film on a Si wafer (see Table S3 in the SI for the process parameters used for this deposition.)

property relationships across synthesis parameter spaces can be mapped accurately and efficiently. Leveraging the autonomous capabilities of the MAP as well as the multi-experiment campaign approach (per substrate), we provide a time-efficient way to explore large parameter spaces. PE-ALD of TiO_2 deposited via our ALDBot tool served as the model system to demonstrate this paradigm, leveraging ALDBot’s full programmatic control and in situ spectroscopic ellipsometry for real-time growth-rate measurements. In a 4D synthesis parameter space, the active learning framework clearly outperformed random sampling in both predictive accuracy and information gain, providing a comprehensive model predicting the growth rate across the whole parameter space. While such complete modeling exceeded our experimental budget in the full 12D parameter space, we demonstrated that even a high-uncertainty model based on a sparsely probed high-dimensional space can provide useful predictions and enable the creation of additional physically interpretable outputs, such as feature-importance analyses. Independent experimental validation of the model’s predictions, along with materials characterization confirming stoichiometric TiO_2 growth, further demonstrated the approach’s ability to create useful models and experimental guidance even under sparse data conditions. The results presented here demonstrate that a fully integrated MAP can be a powerful tool for process development and optimization of gas-phase-deposited functional materials. We believe that the presented framework is straightforward to adapt to other material classes and deposition techniques, such as sputter deposition. It can be viewed as a modular framework that is easily adapted and expanded by selecting the most informative (multi-modal) in situ characterization methods for a given system and/or by incorporating prior (physical) knowledge into the model.

Materials and Methods

Autonomous ALD instrument hardware design

ALDBot was built around a Kimball Physics ultra-high vacuum (UHV) chamber (MCF800-SphSq-G2E4C4A16) with an internal volume of 232.3 cubic inches. A heated sample holder was custom-made from oxygen-free copper and installed in the middle of the deposition chamber. A smaller sample-loading chamber was separated from the deposition chamber by a manual gate valve, facilitating straightforward wafer switching via a manipulator without venting the growth chamber. The chamber pressure was controlled by a throttle/gate valve combination (VAT) that received feedback from an MKS Baratron vacuum gauge with an operational range of 2 mTorr to 2000 mTorr. Additional full-range vacuum gauges (Pfeiffer MPT 200AR and TPR280) were used in both the main chamber and the load lock for monitoring base pressure and venting. The main chamber Pfeiffer gauge was protected from material deposition by a 1 mm flow restrictor (Ideal Vacuum P108654). The chamber was purged using a combination of a Pfeiffer HiPace 300 turbo pump and an Adixen ACP 28G roughing pump, while the load lock was pumped separately by an Adixen Drytel 1025 pump.

A MeaGlow hollow cathode UHV plasma source was attached to the top port of the main chamber to enable plasma-enhanced ALD processes. RF power was supplied by a Seren R301 RF power supply and a Seren MC2 matching unit. Plasma process gas flow (oxygen), ALD dosing line purge (Ar), and chamber purge gases (Ar) were controlled using Brooks GF120 series mass flow controllers (MFCs) with a maximum flow rate of 100 sccm. Every MFC was synchronized with a pneumatic shut-off valve. The timing of the metal-organic precursor dose pulses and purging was controlled using ALD pneumatic valves (Swagelok ALD3). In-situ thickness monitoring was performed with a Film Sense FS8 eight-color ellipsometer. The source and detector modules of the ellipsometer were attached to opposite glass chamber viewing windows at an angle of 71.022° .

Autonomous ALD instrument wiring and control software design

The autonomous ALD process controls are based on a combination of programmable logic controllers (PLCs, Automation Direct Productivity1000 series) and the in-house-developed Python-based experiment control platform, ScopeFoundry [43]. PLCs were used to control all analog-driven and safety-critical parts of the instrument, including activating shut-off and ALD pneumatic valves, setting setpoints and reading MFC process values, performing continuous safety monitoring using the Baratron gauge’s analog pressure readout, and controlling the sample temperature. Furthermore, the duration of precursor dosing and purging was also controlled by the PLCs, since it requires precise timing control on the millisecond time scale. On the device level, the PLC assembly controls were programmed in ladder logic using the vendor software (Productivity Suite). Each internal PLC tag responsible for controlling a single piece of hardware was assigned a Modbus address to interface with higher-level ScopeFoundry control software (see below). ScopeFoundry was used to develop high-level software for the automated ALD control, including a graphical user interface, data management, and automated ML-driven ALD campaign execution. A custom ScopeFoundry

application established communication with every hardware component of the instrument, including the vacuum gauges, the PLC assembly, the ellipsometer, the plasma power supply and matching box, and the throttle valve. An influxdb database was used to log and monitor the status of all instrument components. Furthermore, metadata for every ALD run, including a unique wafer identification number, run parameters, and sample description, was automatically saved with each experiment.

In situ ellipsometry

The ellipsometer was remotely controlled via ScopeFoundry and operated in dynamic measurement mode, enabling continuous in situ monitoring of the sample thickness during each deposition cycle. We used a three-layer model (Si/SiO₂/TiO₂) with fixed parameters for the Si and SiO₂ layers leaving only the thickness and refractive index of the TiO₂ as fit parameters. Here, the refractive index of TiO₂ was constrained to the range 2.0 – 2.5. For every wafer, we used a temperature-dependent two-layer model (Si/SiO₂) to determine the thickness of the native oxide layer before the first deposition. Ellipsometry data fitting was performed in the vendor software, with the model results communicated to the ScopeFoundry app after every deposition cycle. Additionally, the results were confirmed using an independent fitting procedure implemented in Python using the `refellips` package. [44] To extract the growth rate for each experiment, we omitted the first 9 of 25 runs and averaged growth per ALD cycle of the remaining 16 runs.[36] Note that continuous execution of up to 200 experiments per substrate was chosen to enable rapid parameter screening without the need of frequent substrate exchange and conditioning. This procedure should be viewed as a first step to map the parameter space, which subsequently can be accompanied by targeted optimization on individual substrates.

Gaussian process modeling

Gaussian processes were used to model the synthesis-property relationship, mapping the process parameters to growth rate. As a class of non-parametric probabilistic models, Gaussian processes are widely used to approximate unknown latent functions stochastically, yielding both accurate predictions and uncertainty estimates even with relatively small datasets.[30] Gaussian processes inherently decompose uncertainty into aleatoric and epistemic components. Aleatoric uncertainty stems from the natural variability in the collected data and cannot be reduced regardless of the amount of data, quantifying the system’s repeatability. Epistemic uncertainty, on the other hand, arises from limited data and is reducible as more information is acquired. This property can be exploited to actively and sequentially learn the synthesis-property relationship by identifying future experiments that are expected to maximize information gain.[4]

Varying the synthesis parameters affects the growth rate, and our goal in modeling the synthesis-property relationship is to capture how these parameters jointly influence the growth rate. We denote a single combination of parameter values as $\mathbf{x}$ and the space of all acceptable combinations as the synthesis parameter space $\mathcal{X}$, with $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^D$, where D is the number of synthesis parameters. Because the 12 synthesis parameters span disparate physical units and ranges (e.g., seconds for

timing steps, mTorr for pressure, W for RF power), each parameter was min-max normalized to $[0, 1]$ over its experimental range (Table S1/S2) prior to model fitting, placing all input dimensions on a common scale; unless otherwise noted, $\mathbf{x}$ refers to this normalized representation throughout, including as input to the SHAP analysis described below. We consider two cases, with 4- and 12-dimensional parameter spaces (4D and 12D). The experimentally measured growth rate at a given $\mathbf{x} = [p_1 \dots p_{12}]$ is denoted y and treated as a noisy observation of the underlying latent function $f(\mathbf{x})$: $y = f(\mathbf{x}) + \epsilon$, where ϵ represents measurement noise.

A Gaussian process places a prior distribution over the latent function $f(\mathbf{x})$, such that any subset of function values follows a joint multivariate normal distribution. This prior is fully specified by a mean function $\mu(\mathbf{x})$ and a kernel function $k(\mathbf{x}, \mathbf{x}')$. The mean function encodes the expected trend of the latent function f in the parameter space $\mathcal{X}$ in the absence of data. In the absence of prior domain knowledge, we assume an uninformed constant mean function $\mu(\mathbf{x}) = c_1$, where c_1 is a model hyperparameter that is estimated based on the experimental data. The kernel function encodes the covariance between function values at any two points, whether or not they were observed, and controls the smoothness and structure of the learned function. The $\nu = 3/2$ Matérn kernel is used here because it imposes less restrictive smoothness assumptions than the commonly used squared exponential kernel.[45] This kernel function is parameterized by a signal variance σ_s^2 , which controls the overall amplitude, and a lengthscale l , which controls how rapidly the function varies across the parameter space. However, each synthesis parameter may influence the growth rate to varying degrees. Therefore, we use the so-called Automatic Relevance Determination $\nu = 3/2$ Matérn kernel, assigning an independent model length scale l_d to each synthesis parameter dimension d , where $d \in \{1, 2, \dots, D\}$ and D denotes the number of synthesis parameters.[46] We note that the magnitude of l_d can serve as a heuristic indicator of the relative importance of the corresponding parameter in growth rate, with smaller values indicating greater sensitivity of growth rate to the d -th parameter, though we rely on SHAP for the reported importance analysis as it captures feature interactions and yields per-point attributions.[47] Finally, the observation noise ϵ is assumed to be homoscedastic white noise with zero mean and variance σ_n^2 , capturing the data variability. The Gaussian process model is therefore fully described as:

$$f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \quad (1a)$$

$$\epsilon \sim \mathcal{N}(0, \sigma_n^2), \quad (1b)$$

$$y = f(\mathbf{x}) + \epsilon, \quad (1c)$$

$$k(\mathbf{x}, \mathbf{x}') = \sigma_s^2 \left(1 + \sqrt{3} r(\mathbf{x}, \mathbf{x}')\right) \exp\left(-\sqrt{3} r(\mathbf{x}, \mathbf{x}')\right), \quad \text{with} \quad r(\mathbf{x}, \mathbf{x}') = \sqrt{\sum_{d=1}^D \frac{(x_d - x'_d)^2}{l_d^2}}. \quad (1d)$$

The model parameters $\boldsymbol{\theta} = \{c_1, \sigma_n^2, \sigma_s^2, l_1, \dots, l_D\}$, also known as the hyperparameters, can be estimated from the experimental data by maximizing the log marginal likelihood.

After collecting the experimental data, the prior in Eq. 1 is updated to extract the posterior

distribution over the latent function. More specifically, given a training dataset $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$ of N observations, where $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ is the set of observed synthesis parameter combinations and $\mathbf{y} = \{y_i\}_{i=1}^N$ are the corresponding growth rate measurements, the posterior is obtained by conditioning the prior on $\mathcal{D}$. The posterior distribution at a new unobserved point $\mathbf{x}^*$ is also a Gaussian, with mean and predictive variance given below, where the predictive variance $\sigma^2(\mathbf{x}^*)$ quantifies the uncertainty in the predictions:

$$m(\mathbf{x}^*) = \mu(\mathbf{x}^*) + \mathbf{k}_*^T (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} (\mathbf{y} - \boldsymbol{\mu}), \quad (2a)$$

$$\sigma^2(\mathbf{x}^*) = k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}_*^T (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{k}_*, \quad (2b)$$

where $\mathbf{K}$ is the $N \times N$ kernel matrix with entries $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{k}_*$ is the $N \times 1$ vector of covariances between the training points and $\mathbf{x}^*$, and $\boldsymbol{\mu}$ is the vector of prior mean values evaluated at the training points.

SHAP analysis

To interpret the trained Gaussian process model and rank the relative importance of the synthesis parameters, we used SHapley Additive exPlanations (SHAP). [41, 42] SHAP is a model-agnostic interpretability method rooted in cooperative game theory, in which each input feature—here, each synthesis parameter—is treated as a “player” contributing to a “payout,” the model’s prediction for a given input. [48] For a trained model f and an input $\mathbf{x}$ with D features, the Shapley value ϕ_d assigned to feature d represents its average marginal contribution to the prediction, computed by considering all possible orderings in which features are added to a baseline (typically the expected model output over the training data). This procedure guarantees an additive decomposition of the prediction:

$$f(\mathbf{x}) = \phi_0 + \sum_{d=1}^D \phi_d(\mathbf{x}),$$

where ϕ_0 is the baseline (average) prediction and each $\phi_d(\mathbf{x})$ quantifies both the magnitude and direction of the d -th parameter’s influence on the growth rate prediction for that specific input. Because the SHAP decomposition is additive in the model output, each $\phi_d(\mathbf{x})$ carries the same units as the predicted growth rate (Å/cycle). SHAP values are also computed independently for each point in the dataset, allowing them to capture non-linear effects and parameter interactions that a single global sensitivity metric, such as the Gaussian process length scale, cannot resolve. Aggregating the absolute SHAP values across all training points yields a global ranking of parameter importance, while the sign and spread of the per-point values (as visualized in the SHAP summary plot in Figure 3a) reveal whether a given parameter increases or decreases the predicted growth rate across the sampled parameter range.

Model evaluation

To quantify predictive accuracy, we computed the root mean square error (RMSE) between the Gaussian process predictions and held-out experimental data. Because ground-truth growth rates are available only at experimentally measured conditions, the held-out test points were drawn from the pooled experimental data of both the active-learning and random-sampling campaigns (excluding the shared random-initialization experiments—the first 5 in the 4D case and 15 in the 12D case). From this pooled set, we randomly sampled 100 held-out test sets, each containing 50 experiments, using a fixed random seed. Critically, these same 100 test sets were used to evaluate both the active learning and the random sampling models, ensuring that the two approaches are compared on identical held-out data and that neither is tested on a distribution tailored to its own sampling strategy.

For a given campaign (active learning or random) and at each run index i , a Gaussian process model was trained on the first i accumulated experiments of that campaign, with any experiments coinciding with the current held-out test set removed from the training data to prevent data leakage. The trained model then predicted the growth rate at the 50 held-out points, and the RMSE was computed on the original growth-rate scale ($\text{\AA}/\text{cycle}$) as

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2}, \quad (3)$$

where $\hat{y}_i$ is the posterior predictive mean, y_i the measured growth rate, and $N = 50$ the number of held-out test points. This procedure was repeated across all 100 held-out test sets, and the reported RMSE at each run index in Figures 2 and 3 is the mean over the 100 repetitions, with the shaded band in Figures 2c and 3a denoting the standard error of the mean.

Information-theory-based active learning

Identifying the informative experiments through active learning efficiently builds the synthesis-property relationships, especially in multi-dimensional parameter spaces. Traditional parameter space exploration methods include varying one parameter while keeping the others fixed, random sampling, and exhaustive grid-based experimental designs that sample the parameter space $\mathcal{X}$ without leveraging prior knowledge of the system. Such traditional approaches are inherently inefficient and require a large number of experiments to adequately characterize the relationship between synthesis parameters and growth rate, and risk failing to capture parameter interactions.[49] Information-theory-based active learning addresses this by iteratively selecting the most informative experiments based on the current state of knowledge, as captured by the Gaussian process model.[33, 50] This enables the construction of synthesis-property relationships with significantly fewer experiments while naturally accounting for parameter interactions.

Experiments in active learning are selected using an acquisition function, which is usually guided by two competing objectives: exploitation and exploration. Exploitation focuses on sampling regions

where the model predicts desirable outcomes. Exploration, however, focuses on sampling regions of maximum epistemic uncertainty to improve the global understanding of the latent function.[35] The goal here is to efficiently build an accurate synthesis-property relationship across the entire parameter space, $\mathcal{X}$, rather than to find an optimal point. Therefore, we adopt a pure exploration approach, driven by an information-theory-based acquisition function, to maximize the reduction in posterior uncertainty over $\mathcal{X}$. [33]

The acquisition function used here is the total correlation. It identifies the point $\mathbf{x}^* \in \mathcal{X}$ whose function value $\mathbf{f}(\mathbf{x}^*)$ is least correlated with the function values $\mathbf{f}(\mathbf{x})$ at the previously observed points $\mathbf{x}$. [51] Performing experiment at this point is expected to yield the greatest information gain and reduction in epistemic uncertainty across $\mathcal{X}$. Mathematically, total correlation is defined as the Kullback-Leibler (KL) divergence [52] between the joint prior distribution of the noiseless function values where $\mathbf{x}$ and $\mathbf{x}^*$ are correlated, and the joint prior where they are uncorrelated, as expressed in Eq. 4. Minimizing the KL divergence between distributions A and B , therefore, identifies a candidate point that is uncorrelated with the existing observations $\mathbf{x}$, ensuring maximum spatial coverage and information gain across $\mathcal{X}$.

$$\text{KL}\left(p(\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{x}^*)) \parallel p(\mathbf{f}(\mathbf{x})) p(\mathbf{f}(\mathbf{x}^*))\right) = \text{KL}(A \parallel B), \quad (4a)$$

$$A \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}(\mathbf{x}) \\ \boldsymbol{\mu}(\mathbf{x}^*) \end{bmatrix}, \begin{bmatrix} \mathbf{k}(\mathbf{x}, \mathbf{x}) & \mathbf{k}(\mathbf{x}, \mathbf{x}^*) \\ \mathbf{k}(\mathbf{x}^*, \mathbf{x}) & \mathbf{k}(\mathbf{x}^*, \mathbf{x}^*) \end{bmatrix}\right), \quad (4b)$$

$$B \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}(\mathbf{x}) \\ \boldsymbol{\mu}(\mathbf{x}^*) \end{bmatrix}, \begin{bmatrix} \mathbf{k}(\mathbf{x}, \mathbf{x}) & \mathbf{0} \\ \mathbf{0} & \mathbf{k}(\mathbf{x}^*, \mathbf{x}^*) \end{bmatrix}\right). \quad (4c)$$

The active learning procedure used in this study follows a closed-loop sequential design, illustrated in Figure 1b. Starting from a random set of N_0 synthesis parameter combinations to be tested, $\mathbf{x}$, experiments are performed via ALDBot, growth rate measurements are acquired $\mathbf{y} = \{y_i\}_{i=1}^{N_0}$, and the initial dataset $\mathcal{D}$ is defined. Then, a Gaussian process model is fit to the available data, and the total correlation acquisition function is optimized over $\mathcal{X}$ to identify the next most informative point $\mathbf{x}^*$. After that, the experiment is conducted, the resulting growth rate measurement y^* is collected, and the dataset is updated as $\mathcal{D} \leftarrow \mathcal{D} \cup \{\mathbf{x}^*, y^*\}$. The Gaussian process model is then refit on the updated dataset, and the process is repeated until data collection is stopped.

Author contributions

A.G., A.R., S.A., A.M.S., and E.S.B. built the ALDBot system. M.B.A. developed and implemented the active learning framework. T.K. and A.R. carried out all ALD experiments. M.S. performed XPS experiments. M.B.A. and T.K. analyzed results and wrote the first draft of the manuscript. A.M.S. and E.S.B. conceptualized and supervised the overall project. All authors revised the manuscript.

Conflicts of interest

There are no conflicts to declare.

Data availability

The codebase for hardware control, active learning strategy, and data analysis can be found at <https://github.com/MF-ALDBot>.

Acknowledgments

This work was supported by the Laboratory Directed Research and Development Program of Lawrence Berkeley National Laboratory under U.S. Department of Energy contract no. DE-AC02-05CH11231. Work at the Molecular Foundry was supported by the Office of Science, Office of Basic Energy Sciences, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231, under proposal numbers MFP-08961 and MFP-10299.

Supporting Information

ALDBot schematic

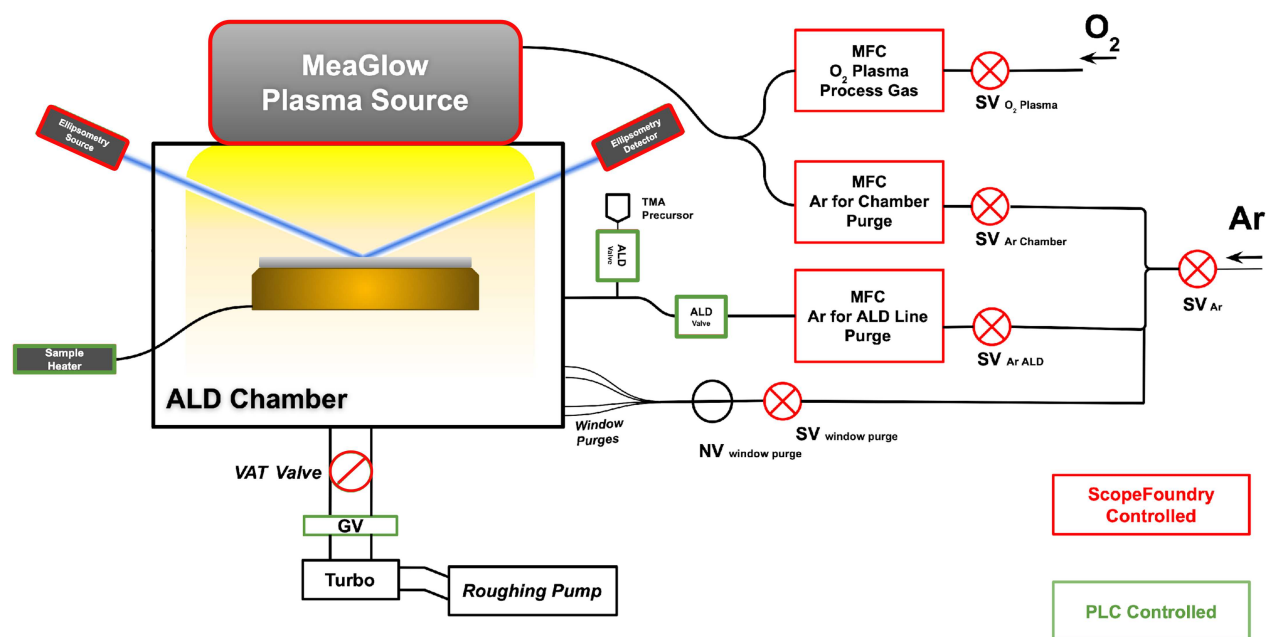


Figure S1. ALDBot schematic showing the gas plumbing and hardware diagram. Frame color indicates where the control logic resides: green for components that are controlled autonomously by the PLCs, red for components sequenced by ScopeFoundry. The PLCs provide the physical I/O for all MFCs and valves, but for every MFC and for all valves except the two ALD valves, they act purely as an execution layer, actuating on command from ScopeFoundry without implementing sequencing logic of their own. ScopeFoundry triggers the ALD valves, but their open, delay and close intervals are timed by PLC ladder logic from presets supplied by ScopeFoundry; likewise, the substrate temperature setpoint is entered through ScopeFoundry while the PID loop lives on the PLC. SV – shut-off valve; NV – needle valve; GV – gate valve; MFC – mass flow controller.

Synthesis parameters of the 4D case

Table S1. ALD synthesis parameter values used in study 1, the 4D case. Parameters shown in blue represent the synthesis parameter space explored, with their respective [Min, Max] limits.

Parameter	Value
Precursor Dose Time p_4 (ms)	[100, 6000]
Precursor Purge Time p_6 (ms)	[10, 6000]
Plasma Preparation Time p_7 (ms)	[10, 4000]
Plasma Purge Time p_{12} (ms)	[10, 6000]
Precursor Preparation Time p_1 (ms)	2000
Process Pressure p_2 (mTorr)	10
Ar Process Flow Rate p_3 (sccm)	50
Precursor Purge Flow Rate p_5 (sccm)	5
O ₂ Plasma Flow Rate p_8 (sccm)	75
Plasma Pressure p_9 (mTorr)	60
Plasma Duration p_{10} (ms)	2000
RF Power Setpoint p_{11} (W)	100

Synthesis-property relationship of the 4D case

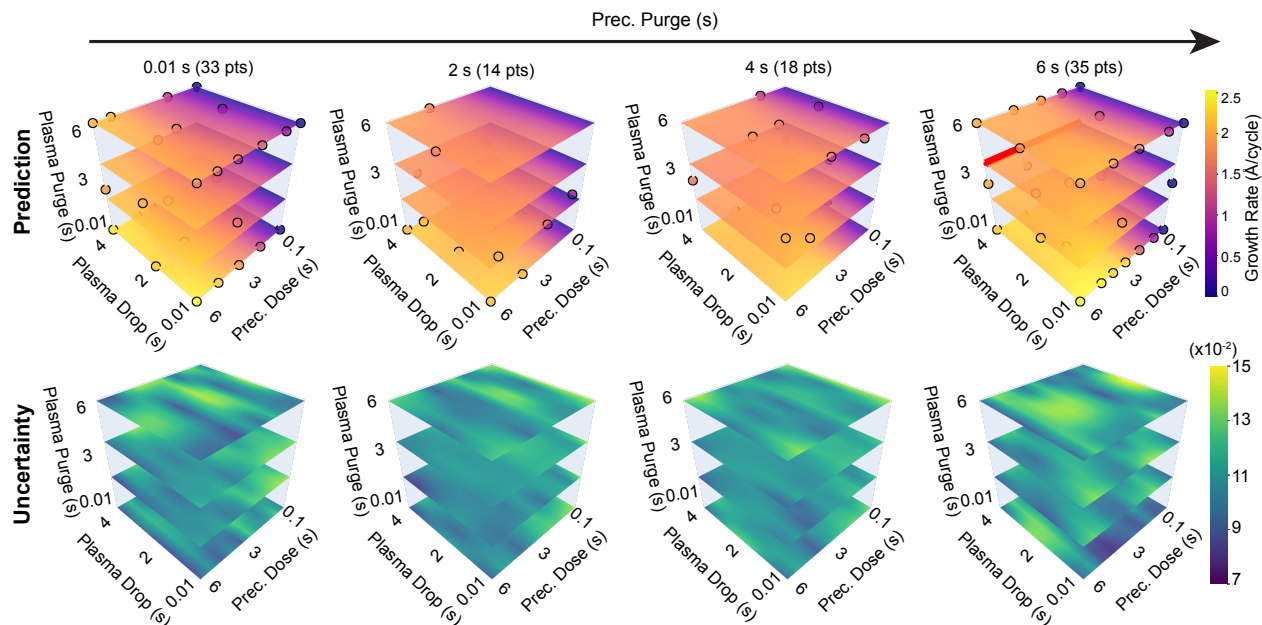


Figure S2. Fully characterized synthesis-property relationship explored via the closed-loop active learning approach for case study 1.

Synthesis parameters of the 12D case

Table S2. ALD synthesis parameter values used in study 2. Parameters shown in blue represent the synthesis parameter space explored, with their respective [Min, Max] limits.

Parameter	Value
Precursor Preparation Time p_1 (ms)	[100, 4000]
Process Pressure p_2 (mTorr)	[5, 300]
Ar Process Flow Rate p_3 (sccm)	[25, 100]
Precursor Dose Time p_4 (ms)	[100, 5000]
Precursor Purge Flow Rate p_5 (sccm)	[1, 20]
Precursor Purge Time p_6 (ms)	[5, 5000]
Plasma Preparation Time p_7 (ms)	[10, 5000]
O ₂ Plasma Flow Rate p_8 (sccm)	[25, 100]
Plasma Pressure p_9 (mTorr)	[25, 300]
Plasma Duration p_{10} (ms)	[100, 5000]
RF Power Setpoint p_{11} (W)	[25, 300]
Plasma Purge Time p_{12} (ms)	[10, 10000]

Un-normalized information gain for the 4D case

Figure S3 shows the unnormalized cumulative information gain, computed as the running sum of the Kullback–Leibler (KL) divergence between successive Gaussian process posterior predictive distributions, for the active learning (blue) and random sampling (red) campaigns, plotted on a logarithmic vertical scale. In absolute terms, the random sampling campaign accumulates substantially more total information gain than the active learning campaign by the end of the run. However, the shape of the two curves reveals markedly different underlying behavior. The random sampling curve is characterized by long, flat plateaus punctuated by abrupt, large jumps, most notably around experiments 37, 48, and 86. These jumps occur when a randomly selected experiment falls into a sparsely sampled region of parameter space where the Gaussian process posterior has been extrapolating with unwarranted confidence; the resulting large KL divergence reflects a substantial corrective update to a previously miscalibrated model, rather than the efficient acquisition of new information. In contrast, the active learning campaign, which explicitly targets regions of high predictive uncertainty at each iteration, avoids these large corrective swings. Its cumulative information gain instead rises smoothly and saturates early, indicating that the model remains well calibrated throughout the campaign and that later experiments contribute comparatively little additional change to the posterior. We note that the absolute magnitude of the KL divergence depends on the Gaussian process kernel and hyperparameters and should therefore be interpreted qualitatively rather than as a model-independent measure of information content. For this reason, the normalized cumulative information gain shown in the main text (Figure 2d) is used to directly compare the two sampling strategies, as it isolates the rate of convergence toward each method’s asymptotic value rather than the raw magnitude of posterior change.

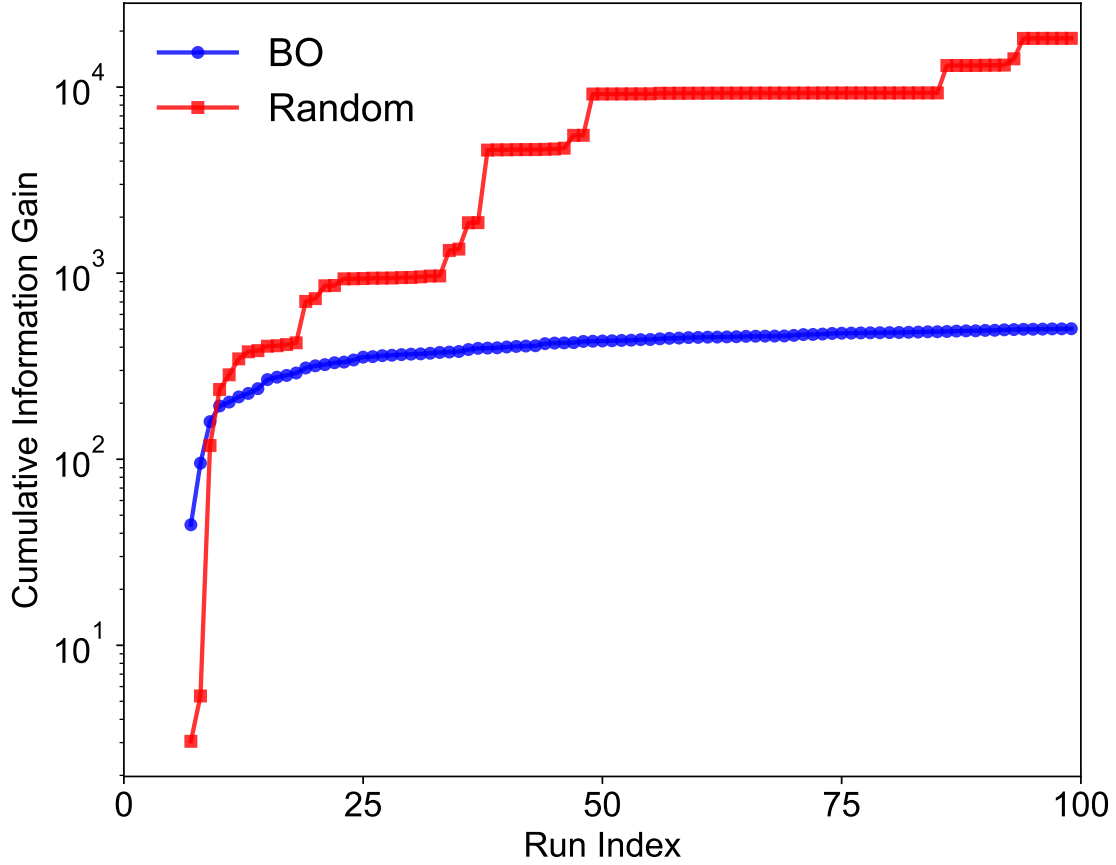


Figure S3. Unnormalized cumulative information gain for active learning and random sampling. Cumulative Kullback–Leibler (KL) divergence between successive Gaussian process posterior predictive distributions as a function of run index for the active learning (blue) and random sampling (red) campaigns, shown on a logarithmic scale. The random sampling campaign exhibits large, discrete jumps in cumulative information gain, corresponding to instances where a randomly chosen experiment falls in an undersampled region and forces a large corrective update to an overconfident GP posterior. The active learning campaign avoids these large corrective jumps by preferentially sampling high-uncertainty regions before miscalibration accumulates, resulting in a smaller-magnitude, smoothly saturating curve. Because absolute KL divergence values are sensitive to the choice of Gaussian process kernel and hyperparameters, these curves should be interpreted qualitatively; normalized versions of these curves (Figure 2d, main text) are used for direct comparison of convergence rates between the two sampling strategies.

Prediction performance comparison in the 12D case

Figure S4 shows the evolution of RMSE for the active learning and random sampling campaigns in the 12D synthesis parameter space, evaluated using the same held-out test procedure described in the Methods. Both curves begin at run index 15, reflecting the shared random-initialization experiments common to both campaigns. Over the first ~ 50 experiments, the two approaches perform comparably, with substantial run-to-run fluctuation in both curves. Beyond this point, the active learning campaign’s mean RMSE consistently falls below that of random sampling, though the improvement is modest compared to the 4D case (Figure 2c): the two approaches’ standard-error bands remain overlapping for essentially the entire campaign, and by the end of the 200-experiment budget, the mean RMSE values of both approaches converge toward similar values as they continue to decrease. Neither curve exhibits the clear plateau observed in the 4D campaign, in which active learning decisively separated from random sampling and stabilized at a lower steady-state RMSE (Figure 2c). This behavior mirrors the early, pre-separation regime of the 4D campaign and suggests that the substantially larger 12-dimensional parameter space requires a correspondingly larger experimental budget before the benefit of active learning becomes fully pronounced, as discussed in the main text as one of its limitations.

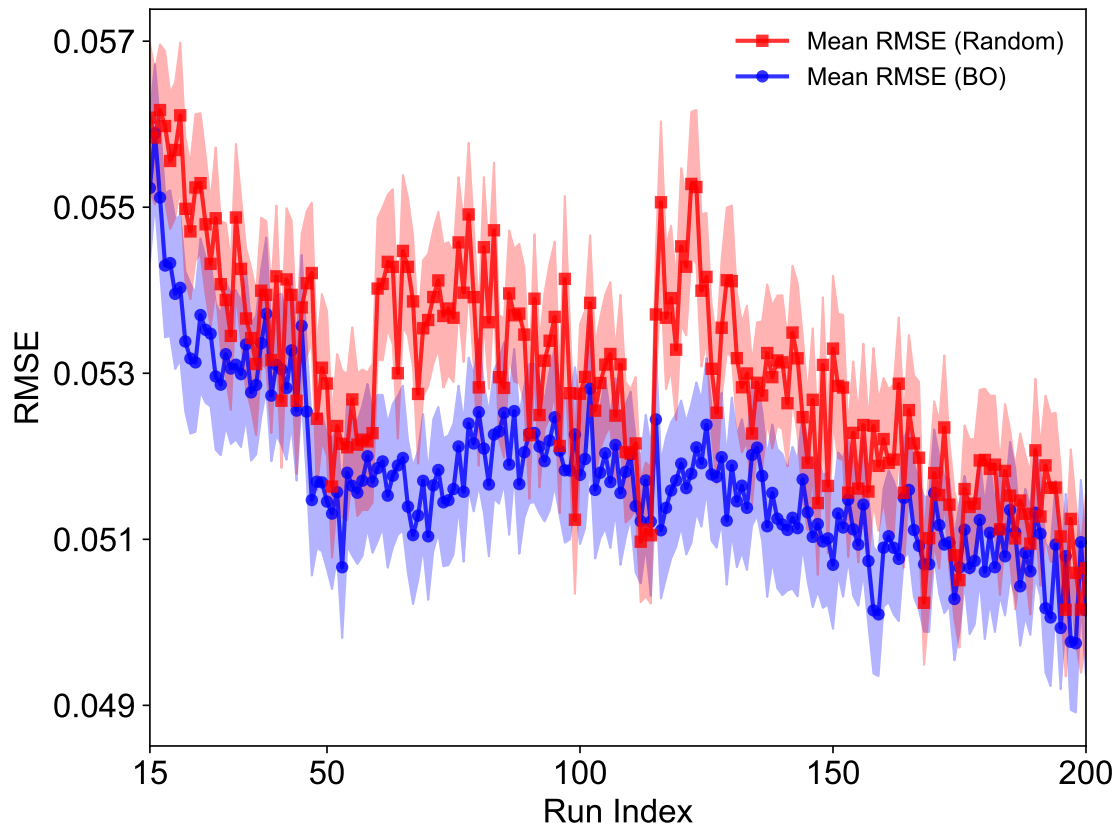


Figure S4. RMSE evolution over the 200-experiment campaign in the 12D synthesis parameter space for the active learning (blue) and random sampling (red) approaches. Markers denote the mean RMSE across repeated held-out test evaluations (see Methods), and shaded regions denote the standard error of the mean. The active learning approach shows a lower mean RMSE than random sampling for most of the run range, particularly beyond approximately run 50; however, the two approaches' standard-error bands overlap throughout the campaign and both curves continue to decrease without reaching a clear plateau by run 200.

Synthesis parameters used for the thin film characterized by XPS

Table S3. ALD synthesis parameter values used in the XPS study.

Parameter	Value
Precursor Preparation Time p_1 (ms)	100
Process Pressure p_2 (mTorr)	5
Ar Process Flow Rate p_3 (sccm)	25
Precursor Dose Time p_4 (ms)	3706
Precursor Purge Flow Rate p_5 (sccm)	14
Precursor Purge Time p_6 (ms)	5
Plasma Preparation Time p_7 (ms)	11
O ₂ Plasma Flow Rate p_8 (sccm)	41
Plasma Pressure p_9 (mTorr)	167
Plasma Duration p_{10} (ms)	2423
RF Power Setpoint p_{11} (W)	299
Plasma Purge Time p_{12} (ms)	10

References

- [1] Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, and Kristin A. Persson. Commentary: The Materials Project: A materials genome approach to accelerating materials innovation. *APL Materials*, 1(1):011002, 2013. ISSN 2166-532X. URL <https://doi.org/10.1063/1.4812323>.
- [2] John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S. Rosen, Gerbrand Ceder, Kristin A. Persson, and Anubhav Jain. Structured information extraction from scientific text with large language models. *Nature Communications*, 15(1):1418, 2024. URL <https://www.nature.com/articles/s41467-024-45563-x>.
- [3] Maged Abdelsamie, Kootak Hong, Kevin Cruse, Christopher J. Bartel, Viktoriia Baibakova, Amalie Trewartha, Anubhav Jain, Gerbrand Ceder, and Carolin M. Sutter-Fella. Combining text mining, in situ characterization, and ab initio calculations to rationalize BiFeO₃ crystallization pathways. *Matter*, 6(12):4291–4305, 2023. URL [https://www.cell.com/matter/abstract/S2590-2385\(23\)00502-7](https://www.cell.com/matter/abstract/S2590-2385(23)00502-7).
- [4] Ansuman Halder, Maher B Alghalayini, Shuan Cheng, Nikil Thalanki, Thong M Nguyen,

- Abigail R Hering, Do-Kyoung Lee, Simon Arnold, Marina S Leite, Edward Barnard, Aleksandr Razumtcev, Morgan Wall, Arian Gashi, Yi-Ru Liu, Marcus M. Noack, Shijing Sun, and Carolin M. Sutter-Fella. Ai-driven robot enables synthesis-property relation prediction for metal halide perovskites in humid atmosphere. *Advanced Energy Materials*, 15(34):2502294, 2025.
- [5] Jiyun Zhang, Jianchang Wu, Vincent M. Le Corre, Jens A. Hauch, Yicheng Zhao, and Christoph J. Brabec. Advancing perovskite photovoltaic technology through machine learning-driven automation. *InfoMat*, 7(5):e70005, 2025. doi: 10.1002/inf2.70005.
- [6] Nathan J. Szymanski, Bernardus Rendy, Yuxing Fei, Rishi E. Kumar, Tanjin He, David Milsted, Matthew J. McDermott, Max Gallant, Ekin Dogus Cubuk, Amil Merchant, Haegyeom Kim, Anubhav Jain, Christopher J. Bartel, Kristin Persson, Yan Zeng, and Gerbrand Ceder. An autonomous laboratory for the accelerated synthesis of inorganic materials. *Nature*, 624(7990):86–91, 2023. ISSN 0028-0836. URL <https://pmc.ncbi.nlm.nih.gov/articles/PMC10700133/>.
- [7] Nastaran Meftahi, Maciej A. Surmiak, Sebastian O. Fuerer, Kevin J. Rietwyk, Jianfeng Lu, Sonia Ruiz Raga, Caria Evans, Monika Michalska, Hao Deng, David P. McMeekin, Tuncay Alan, Doojin Vak, Anthony S. R. Chesman, Andrew J. Christofferson, David A. Winkler, Udo Bach, and Salvy P. Russo. *Advanced Energy Materials*, 13(38):2203859, 2023.
- [8] Martha M. Flores-Leonar, Luis M. Mejía-Mendoza, Andrés Aguilar-Granda, Benjamin Sanchez-Lengeling, Hermann Tribukait, Carlos Amador-Bedolla, and Alán Aspuru-Guzik. Materials Acceleration Platforms: On the way to autonomous experimentation. *Current Opinion in Green and Sustainable Chemistry*, 25:100370, 2020. ISSN 2452-2236. URL <https://www.sciencedirect.com/science/article/pii/S2452223620300596>.
- [9] Abigail R. Hering, Carolin M. Sutter-Fella, and Marina S. Leite. An AI-accelerated pathway for reproducible and stable halide perovskites. *Chemical Society Reviews*, 54(22):10397–10426, 2025. doi: 10.1039/D5CS00715A. URL <https://pubs.rsc.org/en/content/articlelanding/2025/cs/d5cs00715a>.
- [10] Jiyun Zhang, Jens A. Hauch, and Christoph J. Brabec. Toward Self-Driven Autonomous Material and Device Acceleration Platforms (AMADAP) for Emerging Photovoltaics Technologies. *Accounts of Chemical Research*, 57(9):1434–1445, 2024. URL <https://doi.org/10.1021/acs.accounts.4c00095>.
- [11] P. Navabi, R. Ampadi Ramachandran, H. Bhatia, M. Jaber-Douraki, U. Diwekar, C. Sukotjo, and C. G. Takoudis. Artificial intelligence in atomic layer deposition. *J. Vac. Sci. Technol. A*, 43:060801, 2025. doi: 10.1116/6.0004881.
- [12] Gül Dogan, Sinan O Demir, Rico Gutzler, Herbert Gruhn, Cem B Dayan, Umut T Sanli, Christian Silber, Utku Culha, Metin Sitti, Gisela Schütz, et al. Bayesian machine learning for efficient minimization of defects in ald passivation layers. *ACS Applied Materials & Interfaces*, 13(45):54503–54515, 2021.

- [13] Noah H Paulson, Angel Yanguas-Gil, Osama Y Abuomar, and Jeffrey W Elam. Intelligent agents for the optimization of atomic layer deposition. *ACS Applied Materials & Interfaces*, 13(14):17022–17033, 2021.
- [14] Udit Kumar, Ayush Arunachalam, Corbin Feit, S. Novia Berriel, Kanad Basu, Parag Banerjee, and Sudipta Seal. Real-time artificial intelligence enhanced defect engineering in CeO₂ nanostructures. *Journal of Vacuum Science & Technology A*, 41(6):062409, 2023. ISSN 0734-2101. URL <https://doi.org/10.1116/6.0002912>.
- [15] Philipp Haussermann, Nikhil Biju Joseph, and Daniel Hiller. Leveraging Bayesian Optimization Software for Atomic Layer Deposition: Single-Objective Optimization of TiO₂ Layers. *Materials*, 17(20):5019, 2024. ISSN 1996-1944. URL <https://www.mdpi.com/1996-1944/17/20/5019>.
- [16] Sumner B. Harris, Christopher M. Rouleau, Kai Xiao, and Rama K. Vasudevan. Deep learning with plasma plume image sequences for anomaly detection and prediction of growth kinetics during pulsed laser deposition. *npj Computational Materials*, 10(1):105, 2024. ISSN 2057-3960. URL <https://www.nature.com/articles/s41524-024-01275-w>.
- [17] Sumner B. Harris, Arpan Biswas, Seok Joon Yun, Kevin M. Roccapiore, Christopher M. Rouleau, Alexander A. Puretzky, Rama K. Vasudevan, David B. Geohegan, and Kai Xiao. Autonomous Synthesis of Thin Film Materials with Pulsed Laser Deposition Enabled by In Situ Spectroscopy and Automation. *Small Methods*, 8(9):2301763, 2024. ISSN 2366-9608. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/smtd.202301763>.
- [18] Richard W. Johnson, Adam Hultqvist, and Stacey F. Bent. A brief review of atomic layer deposition: from fundamentals to applications. *Materials Today*, 17(5):236–246, 2014. ISSN 1369-7021. URL <https://www.sciencedirect.com/science/article/pii/S1369702114001436>.
- [19] Erwin Kessels, Anjana Devi, Jin-Seong Park, Mikko Ritala, Angel Yanguas-Gil, and Claudia Wiemer. Atomic layer deposition. *Nature Reviews Methods Primers*, 5(1):66, 2025. ISSN 2662-8449. URL <https://www.nature.com/articles/s43586-025-00435-6>.
- [20] Steven M. George. Atomic Layer Deposition: An Overview. *Chemical Reviews*, 110(1):111–131, 2010. URL <https://doi.org/10.1021/cr900056b>.
- [21] Markku Leskelae and Mikko Ritala. Atomic Layer Deposition Chemistry: Recent Developments and Future Challenges. *Angewandte Chemie*, 42(45):5548–5554, 2003.
- [22] Julian Pilz, Alberto Perrotta, Guenther Leising, and Anna M. Coclite. ZnO Thin Films Grown by Plasma-Enhanced Atomic Layer Deposition: Material Properties Within and Outside the "Atomic Layer Deposition Window". *physica status solidi (a)*, 217(8):1900256, 2020.
- [23] Harald B Profijt, SE Potts, MCM Van de Sanden, and WMM Kessels. Plasma-assisted atomic layer deposition: basics, opportunities, and challenges. *Journal of Vacuum Science & Technology A*, 29(5):050801, 2011.

- [24] SE Potts and WMM Kessels. Energy-enhanced atomic layer deposition for more process and precursor versatility. *Coordination Chemistry Reviews*, 257(23-24):3254–3270, 2013.
- [25] Tahsin Faraz, Harm CM Knoop, Marcel A Verheijen, Cristian AA Van Helvoirt, Saurabh Karwal, Akhil Sharma, Vivek Beladiya, Adriana Szeghalmi, Dennis M Hausmann, Jon Henri, Mariadriana Creatore, and Wilhelmus M. M. Kessels. Tuning material properties of oxides and nitrides by substrate biasing during plasma-enhanced atomic layer deposition on planar and 3d substrate topographies. *ACS Applied Materials & Interfaces*, 10(15):13158–13180, 2018.
- [26] Stephan Ratzsch, Ernst-Bernhard Kley, Andreas Tünnermann, and Adriana Szeghalmi. Influence of the oxygen plasma parameters on the atomic layer deposition of titanium dioxide. *Nanotechnology*, 26(2):024003, 2015.
- [27] Samuel Kim, Peter Y Lu, Charlotte Loh, Jamie Smith, Jasper Snoek, and Marin Soljačić. Deep learning for bayesian optimization of scientific problems with high-dimensional structure, 2021.
- [28] Debopriya Ghosh and Javier Cabrera. Enriched random forest for high dimensional genomic data. *IEEE/ACM transactions on computational biology and bioinformatics*, 19(5):2817–2828, 2021.
- [29] Evgenii Tsymbalov, Sergei Makarychev, Alexander Shapeev, and Maxim Panov. Deeper connections between neural networks and gaussian processes speed-up active learning, 2019.
- [30] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.
- [31] Geun Ho Gu, Changhyeok Choi, Yeunhee Lee, Andres B Situmorang, Juhwan Noh, Yong-Hyun Kim, and Yousung Jung. Progress in computational and machine-learning methods for heterogeneous small-molecule activation. *Advanced Materials*, 32(35):1907865, 2020.
- [32] Maher B Alghalayini, Daniel Collins-Wildman, Kenneth Higa, Armina Guevara, Vincent Battaglia, Marcus M Noack, and Stephen J Harris. Machine-learning-based efficient parameter space exploration for energy storage systems. *Cell Reports Physical Science*, 6(4):102543, 2025.
- [33] Philipp Hennig and Christian J Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(6):1809–1837, 2012.
- [34] José M Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. *Advances in neural information processing systems*, 27, 2014.
- [35] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

- [36] Brian C. Welch, Jeanne Casetta, Rajesh Pathak, Jeffrey W. Elam, Celine Pochat-Bohatier, Philippe Miele, and Tamar Segal-Peretz. Atomic layer deposition of nanofilms on porous polymer substrates: Strategies for success. *Journal of Vacuum Science & Technology A*, 43(2): 022402, 2025. ISSN 0734-2101. URL <https://doi.org/10.1116/6.0004187>.
- [37] Marcus Noack, James Sethian, Kevin Yager, Masafumi Fukuto, Petrus Zwart, Daniela Ushizima, Harinarayan Krishnan, Steven Lee, Ronald Pandolfi, Ian Humphrey, David Perryman, Pablo Enfedaque, Alexander Hexemer, Suchismita Sarker, Tobias Weber, Apurva Mehta, Ruipeng Li, Gregory Doerk, and Martin Boehm. Gpcam, 2022. URL <https://doi.org/10.5281/zenodo.21797284>.
- [38] Mohit Malu, Gautam Dasarathy, and Andreas Spanias. Bayesian Optimization in High-Dimensional Spaces: A Brief Survey. In *2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA)*, pages 1–8, 2021. doi: 10.1109/IISA52424.2021.9555522.
- [39] Carl Hvarfner, Erik Orm Hellsten, and Luigi Nardi. Vanilla Bayesian Optimization Performs Great in High Dimensions, 2024.
- [40] Leonard Papenmeier, Matthias Poloczek, and Luigi Nardi. Understanding High-Dimensional Bayesian Optimization, 2025. URL <http://arxiv.org/abs/2502.09198>.
- [41] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.
- [42] Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence*, 2(1):56–67, 2020.
- [43] Daniel B. Durham, D. Frank Ogletree, and Edward S. Barnard. Scanning Auger spectromicroscopy using the ScopeFoundry software platform. *Surface and Interface Analysis*, 50(11): 1174–1179, 2018.
- [44] Hayden Robertson, Isaac J. Gresham, Stuart W. Prescott, Grant B. Webber, Erica J. Wanless, and Andrew Nelson. *refellips*: A Python package for the analysis of variable angle spectroscopic ellipsometry data. *SoftwareX*, 20:101225, 2022. doi: 10.1016/j.softx.2022.101225. URL <https://www.sciencedirect.com/science/article/pii/S2352711022001431>.
- [45] Michael L Stein. *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media, 1999.
- [46] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

- [47] Topi Paananen, Juho Piironen, Michael Riis Andersen, and Aki Vehtari. Variable selection for gaussian processes via sensitivity analysis of the posterior predictive distribution. In *The 22nd international conference on artificial intelligence and statistics*, pages 1743–1752. PMLR, 2019.
- [48] Lloyd S. Shapley. A value for n-person games. In Harold William Kuhn and Albert William Tucker, editors, *Contributions to the Theory of Games, Volume II*, number 28 in Annals of Mathematics Studies, pages 307–317. Princeton University Press, Princeton, NJ, 1953.
- [49] Julia Ling, Maxwell Hutchinson, Erin Antono, Sean Paradiso, and Bryce Meredig. High-dimensional materials and process optimization using data-driven experimental design with well-calibrated uncertainty estimates. *Integrating Materials and Manufacturing Innovation*, 6(3):207–217, 2017.
- [50] David JC MacKay. Information-based objective functions for active data selection. *Neural computation*, 4(4):590–604, 1992.
- [51] Satoshi Watanabe. Information theoretical analysis of multivariate correlation. *IBM Journal of research and development*, 4(1):66–82, 1960.
- [52] Solomon Kullback. *Information theory and statistics*. Courier Corporation, 1997.