

Cite this: *Digital Discovery*, 2026, 5,
1613

Pessimistic asynchronous sampling in high-cost Bayesian optimization

Amanda A. Volk, *^{ab} Kristofer G. Reyes, ^{cd} Jeffrey G. Ethier ^b
and Luke A. Baldwin *^b

Asynchronous Bayesian optimization is a recently implemented technique that allows for parallel operation of experimental systems and disjointed workflows in autonomous experimentation settings. Contrasting with serial Bayesian optimization, which individually selects experiments one at a time after conducting a measurement for each experiment, asynchronous policies sequentially assign multiple experiments before measurements can be taken and evaluates new measurements continually as they are made available. This technique allows for faster data generation and therefore faster optimization of an experimental space. This work extends the capabilities of asynchronous optimization methods beyond prior studies by evaluating policies that incorporate pessimistic and random predictions in the training data set. The conventional realistic prediction method and five additional asynchronous policies were evaluated in a simulated environment and benchmarked with serial sampling. In many of the tested scenarios, the pessimistic prediction asynchronous policy reached optimum experimental conditions in significantly fewer experiments than both existing asynchronous methods and serial policies, and proved to be less susceptible to convergence onto local optima at higher dimensions. Without accounting for the faster sampling rate enabled by asynchronous operation, the pessimistic asynchronous algorithm could result in more efficient algorithm driven optimization of high-cost experimental spaces. Accounting for sampling rate, the presented asynchronous algorithm could facilitate faster and more robust optimization in parallel autonomous experimentation settings.

Received 25th October 2025
Accepted 12th February 2026

DOI: 10.1039/d5dd00477b

rsc.li/digitaldiscovery

Introduction

Asynchronous Bayesian optimization algorithms enable greater flexibility in algorithm assisted experimental workflows. In most traditional scientific experimental procedures, experiments are conducted over distinct process steps.^{1–4} For example, in reaction chemistry research, a reaction process is often conducted in one experimental apparatus, and the synthesized material is characterized with a separate analysis tool. In many scenarios, such as workflows leveraging transmission/scanning electron microscopy, X-ray diffraction, or even high-field nuclear magnetic resonance spectroscopy, the characterization process can take more time than the experiment. For any single experiment, some portion of the equipment is available for use before the completion of the procedure, which means that an algorithm would have the opportunity to select additional tests to run before it has data from the prior experiment.

This issue is further compounded in experimental environments that rely on a human within the experiment conduction and selection loop. Although this limitation is less prevalent in fully closed-loop experimental contexts, many platform designs and experimental systems would benefit from running multiple experiments, or experimental steps, simultaneously.^{5–9}

One strategy for resolving an incomplete utilization of resources is batch sampling, also referred to as parallel sampling. In batch sampling, a set of experiments are defined and conducted with complete utilization of parallelized experimental resources during each stage of an experimental process, then the measurements from that set of experiments are simultaneously returned to the algorithm for selection of the next set of experiments.¹⁰ This approach is suitable for select experimental environments, such as combinatorial screening platforms or high time cost measurements. However, batch sampling poses several intuitive challenges in sampling efficiency. First, while equipment utilization is improved, there is typically still equipment down time when alternating between the different stages of the experiments. Second, batch sampling often does not maximize data availability in algorithm decision making. Unless the experimental system is inherently structured for batch sampling, there is typically a missed opportunity to complete an experiment, and measurement, that informs the

^aNational Research Council, Washington, District of Columbia 20001, USA^bDepartment of Materials Design and Innovation, University at Buffalo, Buffalo, NY 14260, USA. E-mail: luke.baldwin.1@us.af.mil^cMaterials and Manufacturing Directorate, Air Force Research Laboratory, Wright–Patterson Air Force Base, OH 45433, USA^dBrookhaven National Laboratory, Computational and Data Science Directorate, Upton, NY 11973, USA

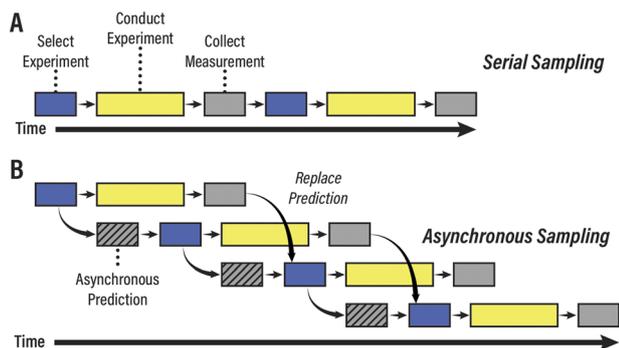


Fig. 1 Illustration of sampling policies that can be harnessed for optimization algorithms. Here a general workflow is depicted for (A) serial sampling and (B) asynchronous sampling.

experiment selection algorithm before conducting all the experiments in the set. Finally, batch methods are not suitable for experimental systems with time dependent outcomes. For example, if an experiment were to produce a material that degrades over time, batch methods would not result in a uniform time step between experiment and measurement, resulting in imprecise data generation.

In response to the constraints of batch sampling strategies, asynchronous sampling methods have recently been implemented in high-cost experimental environments,¹¹ specifically in delocalized experimentation networks.^{12,13} Shown in Fig. 1, asynchronous sampling methods implement similar strategies to batch sampling by selecting multiple experiments without completing measurements. However, in asynchronous designs, experiments are continually measured and added to the data set while other experimental steps are being conducted. In an asynchronous Bayesian optimization design, there is a moving window buffer that contains placeholder data for the currently running experiments. This buffer set is appended to the real value data set for model training. When an experiment measurement is complete, the real data replaces the placeholder data. Then, a new experiment is selected, and the placeholder data is added to the buffer. Several strategies have been implemented to generate placeholder values in asynchronous Bayesian optimization, including local penalty strategies^{14–17} and realistic constant liar predictions,¹¹ among others.^{18,19} Within these studies, several acquisition functions and strategies have been evaluated, including Thompson sampling, expected improvement, and upper confidence bounds. In prior studies, asynchronous sampling resulted in faster data generation rates and therefore faster approach to optimal experimental conditions.

Most prior studies in asynchronous Bayesian optimization algorithms implement a realistic prediction, where the placeholder value is assumed to be the predicted output of the belief model. The assumed primary mechanism of this approach is that the model induces a reduced uncertainty around the previously sampled point, thereby discouraging repeat sampling. In this work, we present five asynchronous sampling value prediction policies that operate around alternative

assumptions: (1) pessimistic constant liar prediction, (2) random prediction, (3) descending pessimism constant liar prediction, (4) ascending pessimism constant liar prediction, and (5) lower confidence bounds prediction. These alternative methods implement three strategies: pessimistic predictions, which presume that the outcome of queued experiments is the most undesired value; random predictions, which presumes a uniform sampling of all possible outcomes without consideration of prior information; and lower confidence bounds predictions, which infers an undesirable outcome within the prediction bounds of the belief model.

In each of these policies, we explore methods for selecting the values used in the placeholder prediction buffer. We benchmark these five policies with serial sampling and realistic prediction asynchronous sampling on a selection of representative ground truth functions using an upper confidence bounds decision policy. It should be noted that other policies could generate alternative outcomes. The simulated optimization campaigns on ground truth functions showed that with an upper confidence bounds decision policy and a Gaussian process regressor, the realistic prediction policy and all five alternative policies outperformed serial sampling considerably when accounting for the improved sampling rate. Furthermore, we found that all five alternative policies consistently performed competitively with serial sampling, and in some cases, significantly outperformed serial sampling when considering the number of experiments conducted. Additionally, the pessimistic policy was shown to provide some durability to low exploration constants in the upper confidence bounds policy, and policy's performance advantage decreases at higher exploration constant values.

The pessimistic constant liar prediction also outperforms serial sampling on discrete input spaces of real-world data modeled with a random forest regressor. The proposed strategies not only generate data at a faster rate than serial sampling, but also select experiments equally or more efficiently. Implementation of the proposed algorithm has notable implications in asynchronous experiment conduction loops for high-cost experiments, and could improve sampling efficiency of serial closed-loop systems. The findings of this study provide a broader context on the role of non-realistic prediction policies in asynchronous Bayesian optimization within real-world relevant design spaces.

Methods

Decision policy

The asynchronous Bayesian optimization simulations were conducted using a framework modified from a previously published library (BOBnd).²⁰ In all simulations, with the exception of the cross-correlation case study, the belief model was a scikit-learn²¹ Gaussian process regressor with a radial basis function kernel and a limited-memory BFGS optimizer. The acquisition function was an upper confidence bounds policy as defined below:

$$x_{\text{Next}} = \operatorname{argmax} \left(y'(x) + \lambda \sigma(x) \right), \quad \lambda = \frac{5}{\sqrt{2}}$$



where x_{Next} is the vector containing the next set of experimental conditions, y' and σ are the model response mean prediction and standard deviation, respectively, as a function of the experimental conditions vector x ; and λ is the exploration constant. The value of λ was selected using a preliminary screening of the serial sampling policy, shown in Fig. S1 in the SI, and was selected for generating reasonable baseline performance on five-dimensional *TriPeak*. It should be noted that selection of this value can significantly impact algorithm performance, a topic discussed in greater detail in the manuscript. Simulations were also carried out with dynamic exploration constants²² on additional ground truth functions from prior literature, shown in SI Fig. S2.

Asynchronous predictions

The asynchronous sampling policy generates a vector of predicted values, referred to as the buffer array (B), for all incomplete experiments. For the simulations conducted in this work, the length of the buffer array (N_{Buff}) varied between one to ten predicted samples, corresponding to one to ten simultaneously running experiments. Each of the six buffer policies – realistic, pessimistic, random, ascending pessimism, descending pessimism, and lower confidence bounds – fill in the buffer arrays B_{Real} , B_{Pess} , B_{Rand} , B_{AscPess} , B_{DesPess} , and B_{LCB} respectively – with the following equations:

$$B_{\text{Real}} = [y'(x_{C+1}), y'(x_{C+2}), \dots, y'(x_{C+N_{\text{Buff}}})]$$

$$B_{\text{Pess}} = [0, 0, \dots, 0]$$

$$B_{\text{AscPess}} = \left[\frac{N_{\text{Buff}} - 1}{N_{\text{Buff}}} y'(x_{C+1}), \frac{N_{\text{Buff}} - 2}{N_{\text{Buff}}} y'(x_{C+2}), \dots, \frac{N_{\text{Buff}} - N_{\text{Buff}}}{N_{\text{Buff}}} y'(x_{C+N_{\text{Buff}}}) \right]$$

$$B_{\text{Rand}} = [u_1, u_2, \dots, u_{N_{\text{Buff}}}] \text{ where } u_i \sim \mathcal{U}(0, 1)$$

$$B_{\text{DecPess}} = \left[\frac{0}{N_{\text{Buff}}} y'(x_{C+1}), \frac{1}{N_{\text{Buff}}} y'(x_{C+2}), \dots, \frac{N_{\text{Buff}}}{N_{\text{Buff}}} y'(x_{C+N_{\text{Buff}}}) \right]$$

$$B_{\text{LCB}} = \left[x_{C_{\text{Buff}}} y'(x_{C+1}) - \lambda \sigma(x_{C+2}), y'(x_{C+2}) - \lambda \sigma(x_{C+2}), \dots, y'(x_{C+N_{\text{Buff}}}) - \lambda \sigma(x_{C+N_{\text{Buff}}}) \right]$$

where x_{C+j} is the input vector for the buffer position (j) after the most recent completed experiment (C), $y'(x)$ is the mean prediction of the belief model at input vector x , and $\mathcal{U}(0, 1)$ is a uniform random distributed that is bounded from 0 to 1. Additionally, all values in x are constrained to the bounds (0, 1). A pessimistic value is defined as the lower bound of the expected response range, which in the case of the *TriPeak* function

is zero. The pessimistic assumption, also referred to as censorship in prior works,²³ has been leveraged in multi-worker contexts where delay distributions are randomly sampled to dynamically determine the buffer lengths, but it has not been evaluated under uniform delay asynchronous sampling. The uniform random distribution is resampled for every value in the buffer each time the model is trained.

Ground truth function

The ground truth function, $f(x)$, referred to as *TriPeak*, is an N -dimensional, triple Gaussian peak integrand function adapted from the BOBnd library,²⁰ Surjanovic and Bingham,²⁴ and Genz,²⁵ and is defined with the equation below:

$$f(x) = c \left(b_1 \exp \left(\sum_{i=1}^D a_1^2 (x_i - \mu_1)^2 \right) + b_2 \exp \left(\sum_{i=1}^D a_2^2 (x_i - \mu_2)^2 \right) + b_3 \exp \left(\sum_{i=1}^D a_3^2 (x_i - \mu_3)^2 \right) \right)$$

$$c = \frac{1}{D(b_1 + b_2 + b_3)},$$

$$a_1 = 4, \quad \mu_1 = 0.2, \quad b_1 = 0.3$$

$$a_2 = 1.5, \quad \mu_2 = 0.5, \quad b_2 = 0.2$$

$$a_3 = 4, \quad \mu_3 = 0.8, \quad b_3 = 0.6$$

where c is the normalization scalar, a_j is the peak width modifier, μ_j is the peak location, and b_j is the peak height modifier for a single dimension of peak j . In the noisy simulations, the ground truth value was sampled with random noise added by sampling from a normal probability distribution with a mean of zero and standard deviation specified by a noise value of 0.01, 0.02, and 0.05 for 1%, 2% and 5% noise, respectively.

Due to the normalization scalar, the function minimum and maximum are equal to 0 and 1, respectively. In the context of these studies, the campaign objective is to maximize the function, and the target feature set (x^*) is defined by:

$$x^* = \text{argmax}(f(x))$$

Simulations using additional ground truth functions are reported in the SI along with randomized sampling control groups, shown in SI Fig. S3. The *TriPeak* function was designed to represent a parameter space that is both reminiscent of real-



world experimental spaces and of non-negligible complexity for algorithm benchmarking. Many common computational benchmarking algorithms pose extreme criteria to navigate, such as many local optima. While relevant in many computational spaces, these functions are far more complex than the surface response typically found in experimental design spaces. Algorithm refinement around these functions, therefore, may not reflect performance in real-world laboratories. Conversely, simple convex unimodal surfaces, while common in experimental optimization spaces, are often not of high enough complexity to justify algorithm development and optimization.

Equivalent experiment time assumption

The simulation results are evaluated with respect to the total number of simulated experiments used in the optimization, referred to as the experiment number, as well as the theoretical optimization time relative to the duration of a simulated experiment, referred to as the time. The optimization time is meant to reflect the improved time cost that could be achieved in a parallel experimental setting. In serial sampling campaigns, the time step for each optimization decision (Δt) is represented as equivalent to the presumed time cost for a single experiment (t_{Exp}). In the asynchronous campaigns, the decision time step is compressed as a function of the policy buffer length, as defined in:

$$\Delta t = \frac{t_{\text{Exp}}}{N_{\text{Buff}} + 1}$$

This compression occurs from the assumptions that multiple experiments are conducted simultaneously, new experiments are executed as soon as the oldest running experiment completes, and all experiments are of equal duration. This compression also assumes that algorithm calculation times and equipment execution limitations are negligible relative to the parallel experiment conduction. While this asynchronous time compression is relevant to many experimental workflows, a simple example would be automated formulation, reaction, and characterization of liquid samples in a well plate using a liquid handler. In a serial configuration, the system would remain idle during the reaction phase of an experiment, then select a new condition after characterizing the complete reaction. In the asynchronous configuration, the system can prepare and/or characterize other experiments during this downtime, thereby increasing throughput.

Loss metric definitions

All algorithms are benchmarked using the median loss metric, and the variability of algorithms are evaluated with the inner quartile range of the loss. The median loss metric (\widetilde{L}_k) and inner quartile range (IQR_k) for a given experiment number (k) are formally defined as:

$$L_{r,k} = f(x^*) - \max([f(x_{1,r}), f(x_{2,r}), \dots, f(x_{k,r})])$$

$$L_k = [L_{k,1}, L_{k,2}, \dots, L_{k,N_{\text{Rep}}}]$$

$$\widetilde{L}_k = \text{med}(L_k)$$

$$\text{IQR}_k = \text{IQR}(L_k)$$

where r is the campaign replicate number, $L_{r,k}$ is the loss value for a given replicate and sample number as calculated from the best sampled conditions so far in the optimization, L_k is the loss vector across all replicates for a given experiment number, and N_{Rep} is the total number of replicates in the campaign. Note that the ground truth function assumes no noise for loss calculations regardless of the noise applied in the campaign.

Results

The five alternative asynchronous sampling policies were designed with the intent of encouraging more robust exploration within the placeholder mechanism. Our assumption with realistic placeholder policies was that the reduction in uncertainty at the sampled point is insufficient for promoting variation in sample selection during asynchrony. As such, we have selected placeholder policies that select deviations from the predicted value. The pessimistic policy, which assumes that all placeholder values are the worst possible outcome, was designed to most dramatically discourage resampling at currently running experimental conditions. The ascending and descending pessimism algorithms are variants of this design where we presume that a dynamic balance between pure pessimism and pure realistic predictions can attain the advantage of pessimism without fully replacing belief model understanding. Similarly, the lower confidence bounds policy was structured so that predictions assume the worst predicted outcome of the belief model, which in theory could preserve the belief model structure while discouraging resampling. Finally, the random value prediction policy assumes that there is no reasonable estimate for the sampled positions and that sampling from a uniform random distribution provides a valid prediction.

Asynchronous pessimistic and random performance

These asynchronous policies were studied with a five-dimensional *TriPeak* ground truth function, with the results shown in Fig. 2 and S4. All five policies demonstrated some viability in accelerating optimization rates through parallel experimentation. However, the realistic prediction policy exhibited significant losses in sampling efficiency when increasing the buffer length to four samples or higher. Additionally, no asynchronous buffer length with a realistic policy outperformed serial sampling when evaluated as a function of the number of experiments. This observation could be attributed to the potential for premature convergence in realistic asynchronous predictions. For example, if the policy samples in a parameter space region with a high predicted reward and low uncertainty, then the realistic policy would be more likely to resample in that region than to explore alternatives in the parameter space. Among the six tested policies, a pure pessimistic policy had the highest, and most consistent,



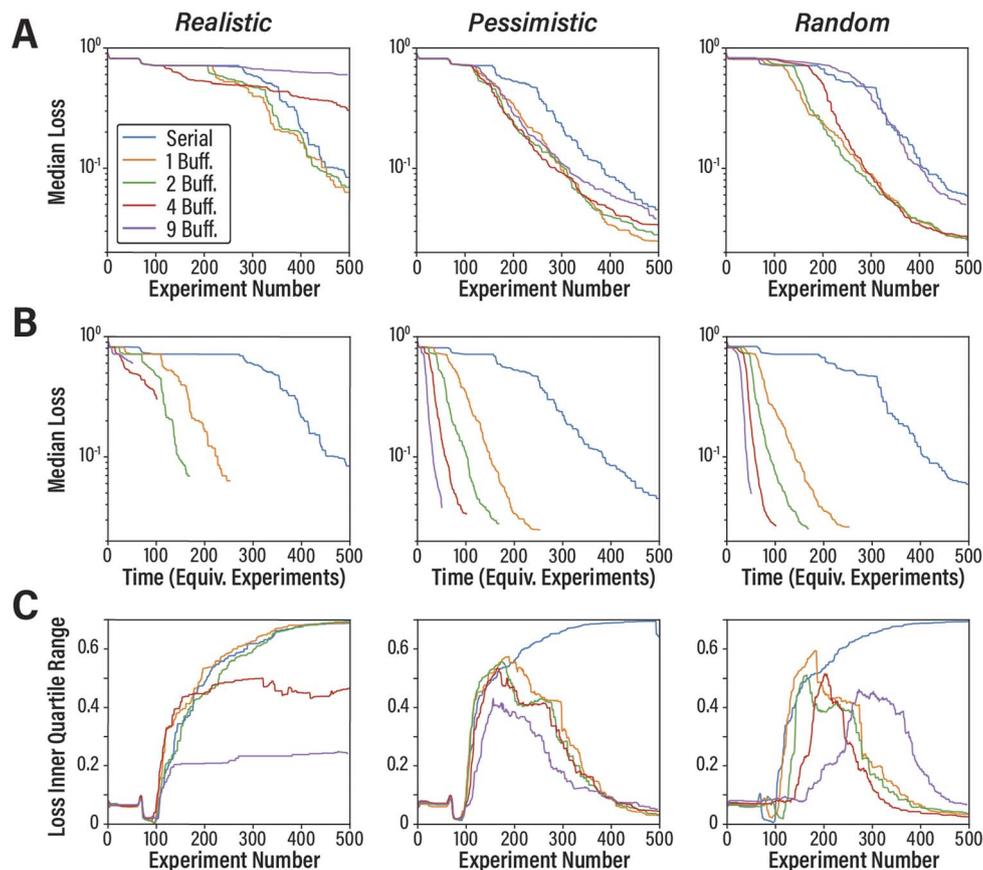


Fig. 2 Simulation results of four asynchronous decision policies utilizing different prediction strategies on five-dimensional *TriPeak*. The median loss across all 200 randomized simulated campaigns as a function of (A) the number of experiments and (B) the effective optimization time relative to a single experiment and (C) the inner quartile range of the loss as a function of experiment number across the four decision policies, (first column) realistic, (second column) pessimistic, and (third column) random. The serial replicates were repeated for each of the policies.

performance across all campaign replicates and buffer lengths, with the descending pessimism performing similarly. While the ascending pessimism policy performed more favorably than the realistic policy across all buffer lengths, the highest buffer length showed some indication of a less consistent, or slower, convergence onto the optimum. Similarly, the random prediction policy showed comparable performance to pessimistic policies, but the optimization efficiency over serial sampling is lost at 9 buffers.

Both the pessimistic and descending pessimism policies outperformed serial sampling for all tested buffer lengths. After 500 experiments, all buffer lengths in the descending pessimism policy reached approximately 60% of the final loss achieved in serial sampling. For the pessimistic policy the 1, 2, 4, and 9 buffer campaigns reached equivalent performance to the serial campaign with 500 experiments in approximately 380, 380, 410, and 480 experiments respectively. After 500 experiments, all asynchronous policies leveraging pessimism showed a significant reduction in the inner quartile range, with the pessimistic policy reaching over an order of magnitude lower inner quartile range than the serial trial. The realistic prediction and serial policy inner quartile ranges continued to increase or plateau after 500 experiments.

The median optimization performance results suggest that the presence of pessimism in asynchronous policies provides more time and material efficient optimizations. Additionally, the narrower convergence in the inner quartile range across trials suggests these results can be achieved more consistently than realistic or serial methods. Despite achieving a lower global accuracy, the pessimistic policy more quickly reaches an accurate estimate of near-optimal conditions than serial policies, shown in SI Fig. S5. Increasing the range of pessimistic prediction policies can further increase the consistency with which campaigns reach optimal conditions. Additionally, the greatest algorithm improvement is observed after the inclusion of a single pessimistic prediction, *i.e.* one buffer length for the pessimistic, ascending pessimism, and descending pessimism policies. Significant improvements are observed with modest additions of pessimistic predictions, while realistic predictions either have no impact or decreased algorithm performance. Shown in SI Fig. S6 and Section S.2, the constant buffer length implementation outperformed a simulated scenario where buffer lengths are changing dynamically between zero and the specified buffer throughout the optimization. Additionally, the prediction of pessimism near predicted optima is shown to be important, as the introduction of randomized pessimism on



a realistic buffer policy performed significantly worse than the pessimistic policy.

Similar trends are observed across both pessimistic and realistic policies when varying the exploration constant across five different values, as shown in SI Fig. S1. Pessimism generates the most significant improvement over serial sampling when lower exploration constants are used, and it influences experimental efficiency less significantly when higher exploration constants are used. For the two lowest value exploration constants tested, the serial policy quickly plateaued and reached a loss approximately 16 times higher than any of the pessimistic buffer policies. For the middle exploration constant value, the serial policy demonstrated improvement with increasing experiments and reached a loss approximately 1.6 times higher than any of the pessimistic policies. However, for the two highest exploration constant values, the serial policy reached a similar loss to the pessimistic policies. The greatest discrepancy occurred between the serial and nine pessimistic buffer policies at the highest exploration constant value, where the serial policy reached a 33% lower loss. The inverse relationship between exploration constant values and the performance advantage of the pessimistic policy suggests that forced exploration may be one of the mechanisms that improve optimization efficiency.

More interestingly, pessimistic policies were more robust to the selection of the exploration term than the realistic policies. Across all tested exploration constant values, the one and two buffer length realistic policies performed similarly to the serial policy. The four and nine buffer length realistic policies, however, demonstrated substantial performance drops, particularly in the scenarios where the serial policy performed well. At the highest exploration constant value, for example, the nine-buffer realistic policy reached an order of magnitude higher loss. Due to the sensitivity of the design space towards the exploration constant, the advantage of pessimism with better tuned hyperparameters is unclear. With this in mind, a logarithmic increasing λ value policy, which in some scenarios outperforms fixed constants, was benchmarked on the pessimistic and realistic policies. Shown in SI Fig. S2, when a more robust and dynamic exploration term is used, the improvement of pessimism over realistic policies is reduced further, indicating that higher exploration rates reduce the relative effectiveness of the pessimistic policy.

Applying pessimism with dynamic exploration terms on high complexity surfaces and very low complexity surfaces generates negligible improvement for most conditions, shown in SI Fig. S2. Using the dynamic exploration term across all ground truth functions, with the one exception of the very simple surface function *Trid*, the pessimistic policy performed either equivalently or better than serial and realistic policies as a function of experiment number. No discernable difference could be identified between serial, realistic, and pessimistic policies for the *Ackley*, *Michalewicz*, and *Schewefel* functions for all buffer lengths, except the nine-buffer realistic policy on *Ackley*. All simulation campaigns, however, achieved poor performance on these complex surface functions, suggesting more detailed analyses may be necessary before drawing

conclusions. As shown in SI Fig. S3, the dynamic exploration term policies performed worse than, or equal to, random sampling for the very low complexity ground truth, *Trid*, and high complexity ground truths, *Michalewicz* and *Schewefel*. In all scenarios where the decision policies outperformed random sampling, the pessimistic policy generated a substantial improvement in performance at the highest tested buffer length. When comparing all buffer lengths by experimental time rather than number, the pessimistic policy significantly outperforms serial sampling.

The lower confidence bounds policy performed equivalently to the pessimistic policy when evaluated over one and two buffer lengths, but the policy appears to converge prematurely and perform worse than the serial policy at high buffer lengths. Small buffer lower confidence bounds policies likely behave similarly to pessimistic policies in that the uncertainty near local optima is high enough to provide a sufficiently pessimistic hallucination. The failure at higher buffer lengths could be attributed to excessively confident models near local optima where clusters of buffer experiments are selected. In this latter case, the policy likely behaves more similarly to the realistic policy and provides insufficient pessimism to encourage exploration.

Effects of dimension and noise

This pessimistic asynchronous method also demonstrates higher performance relative to serial sampling at higher dimensionalities. Shown in Fig. 3, the serial policy outperforms all asynchronous pessimistic policies as a function of experiment number for two, three, and four-dimensional ground truth spaces. These results also show that the asynchronous policies considerably outperformed the serial method for five and six dimensional spaces, as seen in the fourth and fifth columns of Fig. 3. In the six-dimensional space, the serial sampling policy improved very little after 2000 experiments, while the pessimistic asynchronous policies approached optimal conditions. The asynchronous method provided a performance advantage for all five studied dimensions when considered as a function of experimentation time.

One potential explanation for the efficacy of pessimism assisted asynchronous sampling strategies is that the pessimistic predictions reduce the occurrence of premature convergence in upper confidence bounds policies. Forcing a pessimistic prediction on what the current model indicates is the optimal condition prevents resampling at that point, and in cases where replicates already exist outside the buffer, increases model uncertainty at that point which enables improved exploration within the peak. This advantage becomes more dominant when the number of local maxima (*i.e.*, the dimensionality of the *TriPeak* function) increases.

The integration of the pessimistic prediction within the model training data set contrasts with prior pessimistic prediction methods on constant buffer length since these systems implement a penalty region over a defined area around the prior data point. It is possible that these penalty region methods could suffer from the curse of dimensionality as the



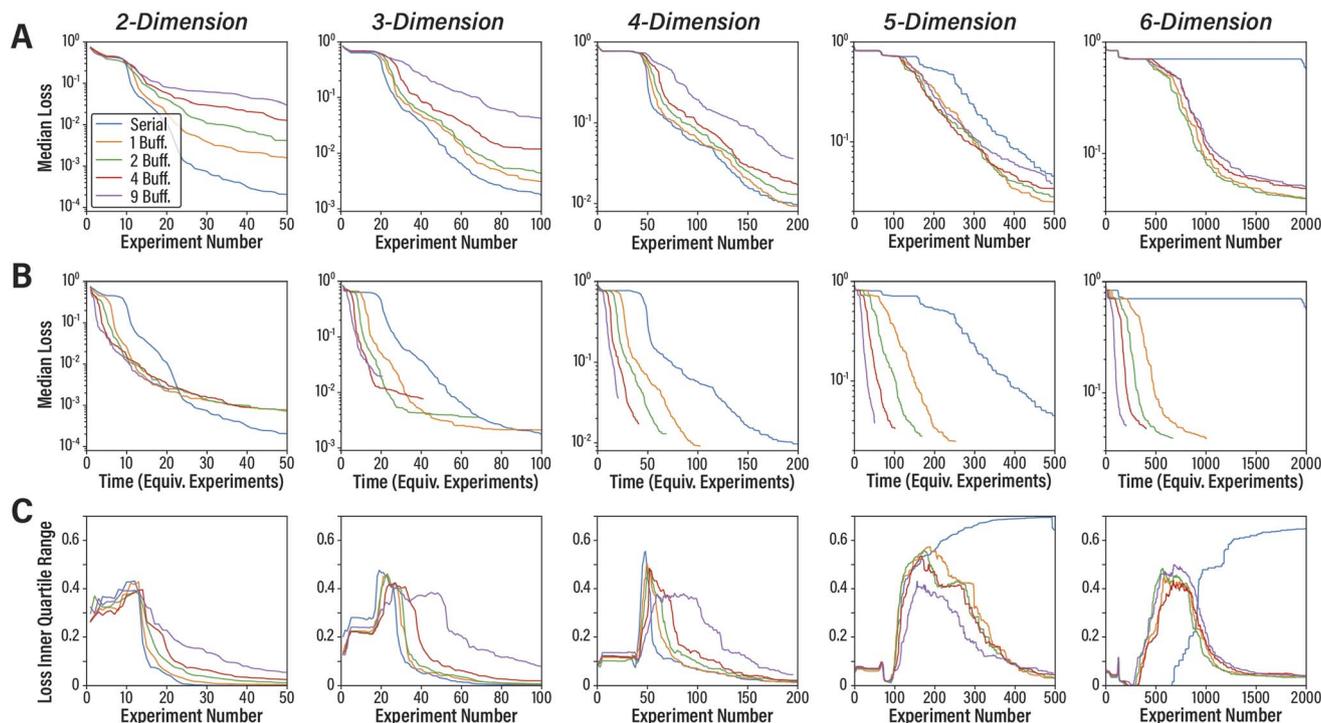


Fig. 3 Simulation results of pessimistic decision policies on *TriPeak* at different dimensionalities and buffer lengths. The median loss across all randomized simulated campaigns as a function of (A) the number of experiments and (B) the effective optimization time relative to a single experiment and (C) the inner quartile range of the loss as a function of experiment number across (columns) two, three, four, five, and six-dimensional ground truth spaces. Each dimensional plot is the result of 200 replicates. The serial replicates were repeated for each of the policies.

volume covered by the defined penalty areas represents a smaller fraction of the overall parameter space.²⁶

A final study was conducted by introducing noise on the five-dimensional *TriPeak* ground truth function using the pessimistic buffer policy across two to six dimensions. Shown in Fig. 4, increasing the noise of the ground truth system resulted in less efficient optimization algorithms in most cases, but the serial policy at higher dimensions gained a performance advantage that is likely due to the normalization effect of noisy sampling. Similar to the no noise simulations, the serial policy outperformed the asynchronous policies for all noise levels at lower dimensionality, and the magnitude of the sampling penalty increased as the buffer size increased. While the introduction of noise negates any advantage with respect to experiment number attained by the buffer policies at five and six dimensions, the asynchronous policies substantially overlap with the results of the serial policy with respect to experiment number at these higher dimensions. The pessimistic policy outperforms the realistic policy for most sets of comparable conditions, as seen in SI Fig. S7. This result further supports the notion that large buffers in pessimistic asynchronous sampling algorithms can provide faster optimizations with negligible impact on experimental efficiency.

By implementing pessimistic predictions through model integration, the asynchronous sampling policies presented here could more effectively navigate higher dimension parameter spaces through more efficient and comprehensive integration of

pessimism. In the complex experimental spaces relevant to algorithm driven experimentation, asynchronous policies provide a notable advantage to serial algorithms when parallel operation is viable. Furthermore, pessimistic asynchronous policies may provide an additional advantage over realistic hallucinations.

Application in real-world cross-coupling reaction data

Finally, the pessimistic asynchronous sampling algorithm was evaluated using a real-world data set centered on a Buchwald–Hartwig cross-coupling reaction space. This reaction system is a common tool in organic chemistry for coupling amines and aryl halides. In a typical design, shown in Fig. 5A, an amine and aryl halide are combined with a base and some additive to perform the coupling. In Ahneman *et al.*²⁷ a database of measured yields for a C–N cross-coupling reaction were generated for 15 aryl halides, 23 additives, 4 catalysts, and 3 bases. This was achieved using a nanomole-scale, 1536-well plate high-throughput experimentation system with yields determined *via* mass spectroscopy. This testing configuration created a complete database for every combination of four categorical features. In total, this database provides a measured yield for approximately 4000 reactions and converted the categorical molecule selection space to a discrete numerical space by providing vector arrays based on molecular descriptors of each of the reactants. This process is discussed in explicit detail in the previously reported work. As a result, this database contains 120 total parameters that go into the yield



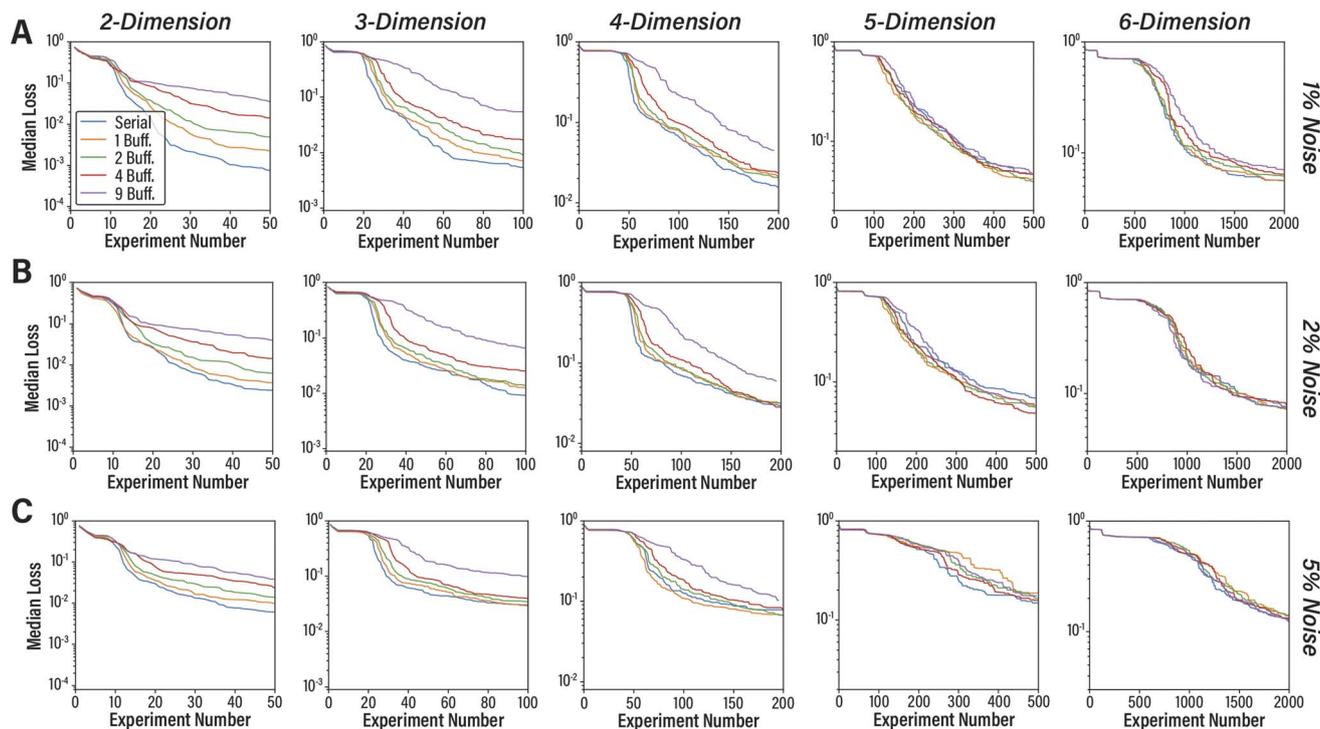


Fig. 4 Simulation results of pessimistic decision policies on *TriPeak* at different dimensionalities and noise levels. The median loss across all randomized simulated campaigns as a function of the number of experiments across (columns) two, three, four, five, and six-dimensional ground truth spaces with (A) 1%, (B) 2%, and (C) 5% noise. Sampling noise is applied by randomly sampling from a normal distribution with a standard deviation equal to the specified noise value and adding the noise sample to the ground truth output. Each dimensional plot is the result of 200 replicates. The loss is calculated from the noiseless ground truth and does not reflect the values sampled from the ground truth during each trials campaign.

measurements. In a simulated campaign, this data was used as a ground truth function for algorithmically optimizing the reaction yield – shown in Fig. 5A.

Like the prior optimization campaigns, the pessimistic asynchronous sampling policy was benchmarked relative to a standard serial optimization strategy. In this specific

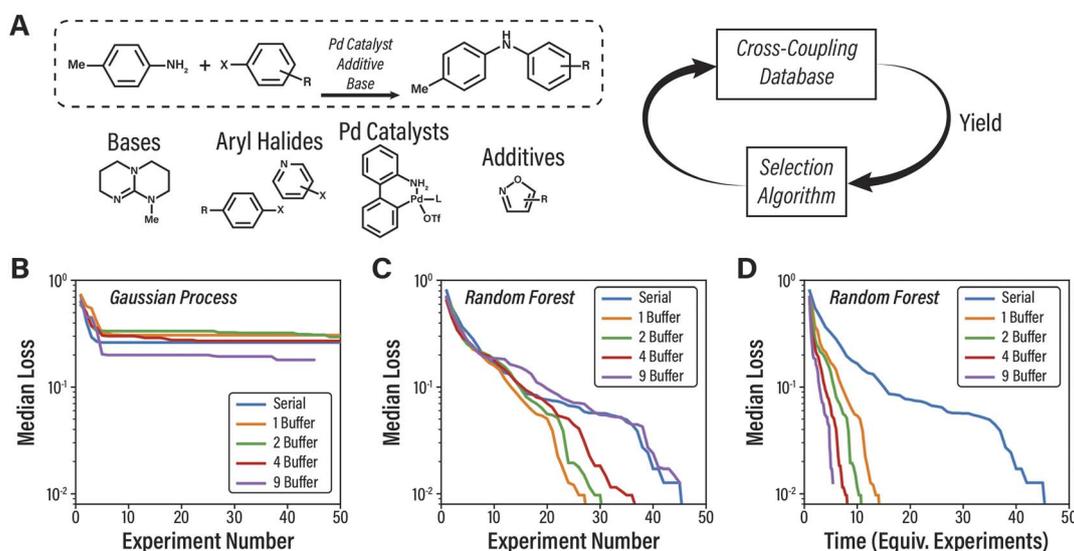


Fig. 5 Asynchronous optimization on real C–N cross-coupling data. (A) Illustration of simulation design setup for sampling from the real-world data set. The median loss across simulated campaigns with the C–N cross-coupling data base on four different pessimistic buffer lengths and serial sampling as a function of (B) the number of experiments and (C) the effective optimization time using a random forest regressor belief model and (D) the number of experiments using a Gaussian process regressor. Each random forest and Gaussian process plot is the result of 90 and 20 replicates respectively. The loss is calculated relative to an assumed maximum yield of 100%.



campaign, we leveraged modeling information from the original study, and the highest performing model from prior work. A random forest regressor was used as the belief model instead of the Gaussian process regressor applied in all prior simulations. The random forest model was selected over Gaussian processes, shown in Fig. 5B, due to the difficulty they exhibited in navigating the experimental space. Uncertainty was estimated through the standard deviation of all forest member predictions.

Of the approximately 4000 samples in the database, two sets of conditions result in a yield of 100%. While the parameter space used for belief model training is technically 120 dimensions, it is highly constrained through discretization of the parameters and a limited number of possible chemical feature combinations. A more realistic approximation of problem space dimensionality is the five categorical parameters. Regardless, pessimistic asynchronous sampling demonstrated a significant improvement over equivalent serial sampling on this system. As seen in Fig. 5C, the pessimistic policy outperformed the serial policy by experiment number for all tested buffer lengths below nine. The highest performing policy, pessimistic sampling with one buffer, reached the optimal yield in 60% fewer samples than serial sampling. While a one sample buffer improved the sampling efficiency over all other methods, increasing the buffer size decreased the efficiency of the policies. The nine-sample buffer performed equivalently to serial sampling with respect to experiment number. Accounting for the accelerated sampling rate of asynchronous policies further amplifies this advantage.

This result not only indicates that asynchronous pessimistic policies can effectively navigate real-world chemistry systems, but it also shows viability in discrete numerical and other constrained spaces. Additionally, the observed improvement over serial methods while using a random forest belief model indicates asynchronous pessimism helps alleviate deficiencies in uncertainty estimation. The use of ensemble member variance as an uncertainty estimator in a non-parametric ensemble would likely not provide an optimal estimator of uncertainty. Despite this, the asynchronous policy performed favorably in a high complexity space. Further exploration and development of these methods could reduce the needs for accurate uncertainty estimates and enable the effective application of different models.

Conclusions

The asynchronous sampling policies presented in this work provide valuable advancements over existing Bayesian optimization strategies for high-cost experimentation in serial experimental systems for simulated and real-world systems. While the pessimistic asynchronous sampling policy suffered a performance penalty in low dimension spaces, it performed considerably better than serial or realistic asynchronous policies in parameter spaces with dimensionalities greater than four. Additionally, the performance gap between serial and pessimistic asynchronous policies appeared to increase as the dimensionality of the parameter space increased. Furthermore,

we have demonstrated that this approach is robust in systems with sampling noise and real-world collected data sets. This work introduces an algorithm specifically designed for the challenges of autonomous experimentation, namely high-dimensional design spaces characterized by significant noise.

Disregarding the increased sampling rate of asynchronous policies, pessimistic policies may offer greater performance for Bayesian optimization algorithms in high-cost sampling systems. When considering the increased sampling rates, pessimistic policies provide a considerable advantage over existing realistic asynchronous and serial sampling approaches. To fully detail the capabilities of the methodologies presented in this work, additional benchmarking studies with similar strategies are required. Further implementation and development of the methods presented here could result in more efficient algorithm driven experimentation and more effective parallelization of experimental processes.

Author contributions

A. V.: conceptualization, data curation, formal analysis, investigation, methodology, validation, visualization, writing – original draft, writing – review and editing. K. R.: validation, writing – review and editing. J. E.: resources, supervision, writing – review and editing. L. B.: conceptualization, funding acquisition, resources, supervision, writing – review and editing.

Conflicts of interest

There are no conflicts to declare.

Data availability

The supplementary information (SI) contains additional figures supporting the main text. The code used to generate the results reported in this manuscript and the accompanying data sets can be found at <https://github.com/Aavolk/BOBnd-Asynch>. (Release <https://doi.org/10.5281/zenodo.18610573>). Supplementary information: additional simulation results (Fig. S.1 to S.7) and asynchronous function descriptions (Section S.1). See DOI: <https://doi.org/10.1039/d5dd00477b>.

Acknowledgements

L. A. B. acknowledges financial support provided by the Laboratory-University Collaboration Initiative (LUCI) Fellowship program from the U.S. Department of Defense Basic Research Office. A. A. V. acknowledge support in part by the appointment to the NRC Research Associateship program at the Air Force Research Laboratory, administered by the Fellowships Office of the National Academies of Sciences, Engineering, and Medicine. K. G. R. was supported by U.S. National Science Foundation award No. 2522770.

References

- 1 A. A. Volk and M. Abolhasani, *Nat. Commun.*, 2024, **15**, 1.



- 2 M. Abolhasani and E. Kumacheva, *Nat. Synth.*, 2023, 2, 483.
- 3 M. Seifrid, R. Pollice, A. Aguilar-Granda, Z. Morgan Chan, K. Hotta, C. T. Ser, J. Vestfrid, T. C. Wu and A. Aspuru-Guzik, *Acc. Chem. Res.*, 2022, 55, 2454.
- 4 J. Park, Y. M. Kim, S. Hong, B. Han, K. T. Nam and Y. Jung, *Matter*, 2023, 6, 677.
- 5 B. Burger, P. M. Maffettone, V. V. Gusev, C. M. Aitchison, Y. Bai, X. Wang, X. Li, B. M. Alston, B. Li, R. Clowes, N. Rankin, B. Harris, R. S. Sprick and A. I. Cooper, *Nature*, 2020, 583, 7815.
- 6 D. Salley, G. Keenan, J. Grizou, A. Sharma, S. Martin and L. Cronin, *Nat. Commun.*, 2020, 11, 1.
- 7 Y. Jiang, D. Salley, A. Sharma, G. Keenan, M. Mullin and L. Cronin, *Sci. Adv.*, 2022, 8, 2626.
- 8 A. E. Gongora, B. Xu, W. Perry, C. Okoye, P. Riley, K. G. Reyes, E. F. Morgan and K. A. Brown, *Sci. Adv.*, 2020, 6, eaaz1708.
- 9 T. Erps, M. Foshey, M. K. Lukovic, W. Shou, H. H. Goetzke, H. Dietsch, K. Stoll, B. Von Vacano and W. Matusik, *Sci. Adv.*, 2021, 7, 7435.
- 10 J. H. Dunlap, J. G. Ethier, A. A. Putnam-Neeb, S. Iyer, S. X. L. Luo, H. Feng, J. A. Garrido Torres, A. G. Doyle, T. M. Swager, R. A. Vaia, P. Mirau, C. A. Crouse and L. A. Baldwin, *Chem. Sci.*, 2023, 14, 8061.
- 11 F. Strieth-Kalthoff, H. Hao, V. Rathore, J. Derasp, T. Gaudin, N. H. Angello, M. Seifrid, E. Trushina, M. Guy, J. Liu, X. Tang, M. Mamada, W. Wang, T. Tsagaantsooj, C. Lavigne, R. Pollice, T. C. Wu, K. Hotta, L. Bodo, S. Li, M. Haddadnia, A. Wolos, R. Roszak, C.-T. Ser, C. Bozal-Ginesta, R. J. Hickman, J. Vestfrid, A. Aguilar-Granda, E. L. Klimareva, R. C. Sigerson, W. Hou, D. Gahler, S. Lach, A. Warzybok, O. Borodin, S. Rohrbach, B. Sanchez-Lengeling, C. Adachi, B. A. Grzybowski, L. Cronin, J. E. Hein, M. D. Burke and A. Aspuru-Guzik, *Science*, 2024, 384, eadk9227.
- 12 J. Bai, S. Mosbach, C. J. Taylor, D. Karan, K. F. Lee, S. D. Rihm, J. Akroyd, A. A. Lapkin and M. Kraft, *Nat. Commun.*, 2024, 15, 1.
- 13 M. Vogler, J. Busk, H. Hajiyani, P. B. Jørgensen, N. Safaei, I. E. Castelli, F. F. Ramirez, J. Carlsson, G. Pizzi, S. Clark, F. Hanke, A. Bhowmik and H. S. Stein, *Matter*, 2023, 6, 2647.
- 14 J. Gonzalez, Z. Dai, P. Hennig and N. Lawrence, in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, PMLR, 2016, pp. 648–657.
- 15 A. S. Alvi, B. Ru, J.-P. Calliess, S. J. Roberts and M. A. Osborne, in *Proceedings of the 36th International Conference on Machine Learning*, PMLR, 2019, pp. 253–262.
- 16 J. P. Folch, R. M. Lee, B. Shafei, D. Walz, C. Tsay, M. Van Der Wilk and R. Misener, *Comput. Chem. Eng.*, 2023, 172, 108194.
- 17 S. Takeno, H. Fukuoka, Y. Tsukada, T. Koyama, M. Shiga, I. Takeuchi and M. Karasuyama, in *Proceedings of the 37th International Conference on Machine Learning*, PMLR, 2020, pp. 9334–9345.
- 18 D. Eriksson, U. Ai, M. Pearce, J. R. Gardner, R. Turner and M. Poloczek, in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019, pp. 5496–5507.
- 19 J. Miguel Hernández-Lobato, J. Requeima, E. O. Pyzer-Knapp, A. Aspuru-Guzik, J. M. Hernández-Lobato, J. Requeima, E. O. Pyzer-Knapp and A. Aspuru-Guzik, in *Proceedings of the 34th International Conference on Machine Learning*, PMLR, 2017, pp. 1470–1479.
- 20 A. Volk and R. Epps, “Modular Bayesian Optimization Benchmarking with n -dimensional functions (BOBnd)” (v1.0.0-alpha), 2024, DOI: [10.5281/zenodo.10644703](https://doi.org/10.5281/zenodo.10644703).
- 21 F. Pedregosa, V. Michel, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, J. Vanderplas, D. Cournapeau, F. Pedregosa, G. Varoquaux, A. Gramfort, B. Thirion, O. Grisel, V. Dubourg, A. Passos, M. Brucher and É. Duchesnay, *J. Mach. Learn. Res.*, 2011, 12, 2825.
- 22 N. Srinivas, A. Krause and M. Seeger, in *Proceedings of the 27th International Conference on Machine Learning*, 2010.
- 23 A. Verma, Z. Dai, B. Kian and H. Low, in *Proceedings of the 39th International Conference on Machine Learning*, PMLR, 2022, pp. 22145–22167.
- 24 S. Surfanovic and D. Bingham, *Virtual Library of Simulation Experiments: Test Functions and Datasets*, 2013, can be found under <https://www.sfu.ca/~ssurjano>.
- 25 A. Genz, in *Proceedings of International Conference on Tools, Methods and Languages for Scientific and Engineering Computation*, Elsevier North-Holland, Inc, 1984, pp. 81–94.
- 26 N. Altman and M. Krzywinski, *Nat. Methods*, 2018, 15, 399.
- 27 D. T. Ahneman, J. G. Estrada, S. Lin, S. D. Dreher and A. G. Doyle, *Science*, 2018, 360, 186.

