Standard Acquisition Is Sufficient for Asynchronous Bayesian Optimization

Ben Riegler
Technical University of Munich
Helmholtz AI
MCML
ben.riegler@tum.de

James Odgers
UTN
Helmholtz AI
MCML

Vincent Fortuin
UTN
Helmholtz AI
MCML

Abstract

Asynchronous Bayesian optimization is widely used for gradient-free optimization in domains with independent parallel experiments and varying evaluation times. Existing methods posit that standard acquisitions lead to redundant and repeated queries, proposing complex solutions to enforce diversity in queries. Challenging this fundamental premise, we show that methods, like the Upper Confidence Bound, can in fact achieve theoretical guarantees essentially equivalent to those of sequential Thompson sampling. A conceptual analysis of asynchronous Bayesian optimization reveals that existing works neglect intermediate posterior updates, which we find to be generally sufficient to avoid redundant queries. Further investigation shows that by penalizing busy locations, diversity-enforcing methods can over-explore in asynchronous settings, reducing their performance. Our extensive experiments demonstrate that simple standard acquisition functions match or outperform purpose-built asynchronous methods across synthetic and real-world tasks.

1 INTRODUCTION

A common problem in many scientific, machine learning, and engineering applications is optimizing functions, which may be evaluated point-wise via running an experiment, but for which there is no known analytical form and no gradient information. Such tasks are known as black-box optimization problems and are often approached using Bayesian optimization (BO). Applications include materials discovery (Packwood et al., 2017; Cinquin et al., 2025), chemical design (Griffiths and Hernández-Lobato, 2020; Wang and Dowling, 2022), as well as nuclear and accelerator physics (Roussel et al., 2024; Ekström et al., 2019). The main application in machine learning is hyperparameter tuning (Chen et al., 2018; Snoek et al., 2012; Wu et al., 2019; Klein et al., 2017).

In practice, it is often possible to run many experiments in parallel. These might correspond to different devices in a wet lab or different GPUs on a compute cluster. Due to varying function evaluation times, practitioners may opt for an asynchronous BO approach to minimize idle times (Zhang et al., 2020; Tran et al., 2022; Egelé et al., 2023; Koyama and Goto, 2022; Frisby et al., 2021). In asynchronous BO, whenever a worker becomes available, a new input location must be selected without access to the outcomes of pending experiments. This challenge leads to the following, seemingly reasonable, hypothesis.

Hypothesis 1 (Standard acquisition is wasteful in asynchronous BO)**.** *Due to the unknown pending evaluations, it is necessary to explicitly enforce diversity in asynchronous BO queries. Standard acquisition functions' failure to do so will result in repeated or redundant queries, wasting evaluation resources, thus leading to poor BO results.*

All existing works on asynchronous BO—explicitly or implicitly—build on Hypothesis 1, by proposing acquisition rules enforcing diversity in the asynchronous queries. In this work, we perform the first critical examination of Hypothesis 1, both conceptually and empirically. We argue that the reasoning in Hypothesis 1 is false, and by extension, proposed methods based on it may be unnecessary.

Instead, we propose to use existing standard acquisi-

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

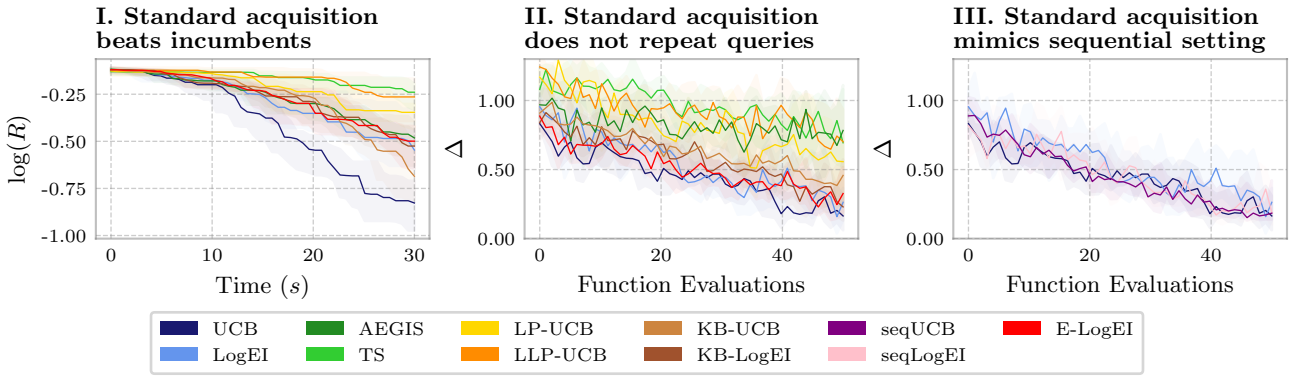

Figure 1: We find that (I.) standard acquisition outperforms or matches purpose-built methods for asynchronous Bayesian optimization. (II.) Comparing the distances of queries to the closest busy location, we see that standard acquisition exhibits the desirable transition from exploration to exploitation, but does not repeat queries. (III.) Standard acquisition functions query at similar distances as their optimally informed sequential counterparts, suggesting that nearby sampling is a desirable feature and should not be prohibited. See Section 2 for a description of the methods and Section 5.2 for a formal definition of the distance Δ. Results are shown on Ackley ($d = 10, q = 8$).

tion rules, such as the Upper Confidence Bound (UCB) (Auer, 2002; Srinivas et al., 2010) or Expected Improvement (EI) (Močkus, 1975). We motivate this theoretically by bounding the regret of the asynchronous standard UCB algorithm. Intuitively, we argue that Hypothesis 1 fails to account for the update to the surrogate model with a new datum, which provides sufficient diversity in asynchronous updates.

Empirically, we show excellent performance on an extensive suite of experiments, supporting our theoretical and conceptual insights regarding the inadequacy of Hypothesis 1. This is further investigated in an analysis of the distances of asynchronous queries to currently busy locations, revealing a similar exploration-exploitation trade-off as that in the optimally informed sequential BO.

Our contributions can be summarized as follows:

- By bounding the Bayes simple regret, we show that the standard UCB in asynchronous BO achieves theoretical guarantees essentially equivalent to those of sequential Thompson sampling.
- We identify a conceptual flaw in the reasoning of Hypothesis 1, as it fails to account for information in the newly available datum before making the next asynchronous acquisition.
- We demonstrate that standard methods, such as the UCB or EI, in asynchronous BO match or outperform purpose-built ones (Figure 1, I.), on synthetic and real-world optimization tasks, providing clear evidence against Hypothesis 1.
- Contrary to Hypothesis 1, we show empirically that standard approaches do not query at and rarely close to currently busy locations (Figure 1, II.), and present evidence that it may not be desirable to explicitly enforce diversity at all (Figure 1, III.).

2 PRELIMINARIES

Formal problem statement In this work, we consider the global optimization of real-valued functions, $f : \mathcal{X} \mapsto \mathbb{R}$, on some compact domain $\mathcal{X} \subseteq \mathbb{R}^d$. It is assumed that $f(\cdot)$ may be queried at some point in the input space $x \in \mathcal{X}$, resulting in a time-delayed observation of the corresponding output $y \in \mathbb{R}$. The evaluation time varies throughout the input space. After n completed evaluations, $\mathbf{y} = \{y_i\}_{i=1}^n$, at inputs $X = \{x_i\}_{i=1}^n$, we have data D_n. In the asynchronous setting with q workers, the unknown function values at busy locations, $\mathcal{B} = \{x_j\}_{j=1}^{q-1}$, are denoted by $\mathbf{y}_b = \{y_j\}_{j=1}^{q-1}$, which we collect in unobserved data D_b.

Bayesian optimization Designed for expensive-to-evaluate black-box objective functions, Bayesian optimization is a sample-efficient global optimization framework (Jones et al., 1998). In order to optimize the objective, $f(\cdot)$, with mere point-wise evaluation, Bayesian optimization represents the uncertainty in the objective via a probabilistic surrogate model, built from D_n. The surrogate then informs an acquisition function, $\alpha : \mathcal{X} \mapsto \mathbb{R}$, proposing the next query location $x' \in \mathcal{X}$.

See Figure 2 for an illustration of synchronous vs. asynchronous BO and Algorithm 1 for pseudo code of asyn-

Algorithm 1 Asynchronous BO with q workers

Require: Oracle $f(\cdot)$, acquisition function $\alpha(\cdot)$, globally stored data $D_0 = \{(x_i, y_i)\}_{i=1}^{n_0}$, time budget T

initialize: $\mathcal{B} \leftarrow \{x_j\}_{j=1}^{q}$ quasi-random
spawn: q worker processes
start timer
for $x_j \in \mathcal{B}$ **do**
  start worker j with query $f(x_j)$
end for
while elapsed time $< T$ **do**
  if j completes query (x_{n+1}, y_{n+1}) **then**
    $D_{n+1} \leftarrow D_n \cup \{(x_{n+1}, y_{n+1})\}$
    $\mathcal{B} \leftarrow \mathcal{B} \setminus \{x_{n+1}\}$
    $\mathcal{GP}_j \leftarrow$ fit-surrogate$(\mathcal{GP}_j, D_{n+1})$
    $x' \leftarrow \arg\max_{x \in \mathcal{X}} \alpha(x \mid D_{n+1})$
    $\mathcal{B} \leftarrow \mathcal{B} \cup \{x'\}$
    start worker j with query $f(x')$
  end if
end while
return $i^* = \arg\max_i y_i$ and (x_{i^*}, y_{i^*})

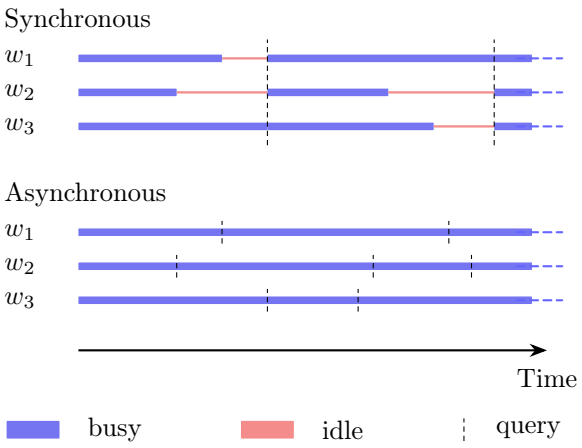

Figure 2: Synchronous vs. asynchronous BO with $q = 3$ workers. The asynchronous BO can run more experiments in the same overall time.

chronous BO with q workers. Importantly, in this work we consider a fully distributed set up, where each worker performs all steps of the BO algorithm locally, with access to the globally shared data set, D_n. Thus, the only source of idle workers is reading from and writing to D_n. These operations are protected by a lock, and the resulting idle times are negligible in practice.

Gaussian Process surrogate A zero-mean Gaussian Process (GP) prior, $\mathcal{GP}(0, k_\phi(\cdot,\cdot))$, is placed on $f(\cdot)$ and defined through a positive definite covariance function $k_\phi : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$, with hyperparameters ϕ, including lengthscales $\ell \in \mathbb{R}^d$. The GP is a standard choice, as it allows for the incorporation of prior knowledge on smoothness, periodicity, and trend of $f(\cdot)$, as well as an analytical posterior (Williams and Rasmussen, 2006).

For a Gaussian observation model with noise η^2 and data $D_n = (X, \mathbf{y})$, the posterior predictive distribution at any given input location, x, is Gaussian with,

$$p(f(x) \mid D_n) = \mathcal{N}(f(x) \mid \mu(x \mid D_n), \sigma^2(x \mid D_n)). \quad (1)$$

The posterior mean and variance are analytically tractable as,

$$\mu(x \mid D_n) = k_\phi(x, X) M_{XX} \mathbf{y} \quad (2)$$
$$\sigma^2(x \mid D_n) = k_\phi(x, x) - k_\phi(x, X) M_{XX} k_\phi(X, x) \quad (3)$$

with $M_{XX} \equiv (K_{XX} + \eta^2 \mathbf{I}_n)^{-1}$ and K_{XX} the kernel matrix of input locations X, i.e., $[K_{XX}]_{ij} = k_\phi(x_i, x_j) \; \forall \; i, j \in [n]$.

Acquisition functions in asynchronous BO In sequential and asynchronous BO alike, a new input location $x' \in \mathcal{X}$ is chosen, as soon as an evaluation resource (e.g., a GPU) becomes available. The choice, x', is made by maximizing the acquisition function.

Standard acquisition functions considered in this work are the Upper Confidence Bound (UCB) (Auer, 2002; Srinivas et al., 2010) and the Expected Improvement (EI) (Močkus, 1975; Jones et al., 1998). For numerical reasons (Ament et al., 2023) we work with the natural logarithm of the EI (LogEI). Due to the insistence on Hypothesis 1, to the best of our knowledge, these standard heuristics have not been considered in previous literature, not even as simple baselines. Instead, numerous works have devised elaborate solutions, which can be grouped as follows.

In asynchronous BO, *Monte Carlo* sampling methods, such as the expected LogEI (E-LogEI), estimate the expected acquisition function given $q-1$ busy locations, $\mathcal{B} \in \mathcal{X}^{q-1}$, under the surrogate (Ginsbourger et al., 2011; Janusevskis et al., 2012; Snoek et al., 2012). *Hallucination* heuristics, such as the Kriging Believer (KB), replace the $q-1$ unknown function values with some value, e.g., the posterior mean (Ginsbourger et al., 2010, 2007). *Randomness*-based heuristics, e.g., Thompson sampling (TS) or AEGIS, introduce a sampling step in the acquisition, aiming to ensure query diversity (Kandasamy et al., 2018; De Ath et al., 2021b). *Penalization*-based methods (LP and LLP) down-weight the acquisition function at and around busy locations (Alvi et al., 2019). A more detailed discussion of these methods can be found in Appendix A.

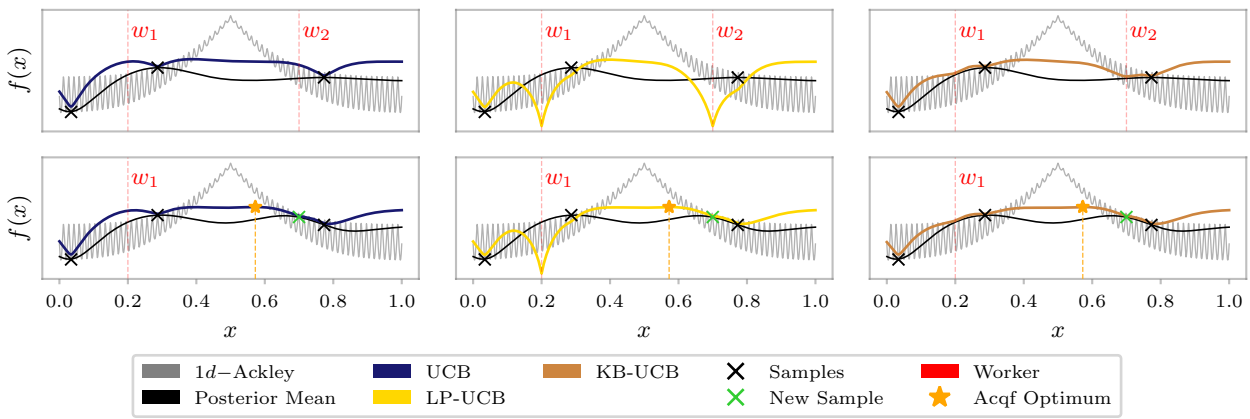

Figure 3: One full asynchronous BO step for the UCB (left), LP-UCB (middle), and the KB-UCB (right). Row 1 corresponds to line 6 in Algorithm 1, with $n_0 = 3$ initial samples and $q = 2$ initialized workers, w_1 and w_2. In row 2, w_2 finishes first, the GP-surrogate is updated with the new sample (green), and the acquisition function is optimized (orange). Notably, all methods query almost the same location, even the standard UCB, which does not take into account the busy locations. See Figure E.1 for subsequent iterations showing a similar trend.

3 THEORETICAL ANALYSIS

The prior works presented in Section 2, concerned with resolving Hypothesis 1, do not provide any theoretical or empirical evidence for repeated or redundant queries, despite using them as a motivation for their methods. In this section, we present theoretical arguments for why standard acquisition can work well in asynchronous BO and why certain purpose-built methods might not.

3.1 Bayes simple regret bound for asynchronous UCB

While not considering busy locations, $\mathcal{B}$, in the acquisition may seem counter-intuitive, this is a commonality of our approach with work on asynchronous TS by Kandasamy et al. (2018). Perhaps surprisingly, the authors show that asynchronous TS achieves essentially equivalent theoretical performance for a given number of observations, n, as its optimally informed sequential counterpart. This is supported by showing that the respective Bayes simple regrets (BSR) after n completed evaluations,

$$\mathrm{BSR}(n) = \mathbb{E}[f(x^*) - \max_{i \in [n]} f(x_i)], \tag{4}$$

admit upper bounds of asymptotically the same order, $\mathcal{O}(\cdot)$, in n. The expectation is w.r.t. the prior on f, the observation noises $\{\epsilon_i\}_{i=1}^n$, as well as any randomness in the algorithm, e.g., GP posterior samples in TS.

For the asynchronous UCB, we present the following bound on the BSR, which is equivalent to the bound on asynchronous TS in Kandasamy et al. (2018).

Theorem 1 (Informal. Bound on BSR(n) for asynchronous UCB)**.** *Let $f \sim \mathcal{GP}(0, k_\phi(\cdot,\cdot))$. Then, for the UCB acquisition function used asynchronously (see Algorithm 1), the Bayes simple regret after n queries can be bounded as*

$$\mathrm{BSR}(n) \lesssim \sqrt{\frac{\xi_q \log(n) \Psi_n}{n}}. \tag{5}$$

Proof. A formal statement and proof are given in Appendix B. □

The bound is given in terms of the maximum information gain (MIG), Ψ_n, after n evaluations, and the quantity ξ_q. Under an RBF kernel, the MIG grows as $\Psi_n \propto \log(n)^{d+1}$ (Srinivas et al., 2010), i.e., sub-linearly. Intuitively, ξ_q represents the cost of operating asynchronously, quantifying the missing information in the $q-1$ busy locations, and is bounded independently of n. See Appendix B for formal definitions of both Ψ_n and ξ_q.

By the same reasoning as in Kandasamy et al. (2018), this provides strong theoretical motivation for why the standard UCB should work well in the asynchronous setting. Namely, its BSR(n) admits an upper bound of the same asymptotic order as that of the optimally informed sequential TS. In particular, this shows that good performance can be theoretically guaranteed, even when neither accounting for busy locations nor injecting randomness. Further intricacies, including the bounds presented by Kandasamy et al. (2018) are discussed in Appendix B.

3.2 The expected UCB is the Kriging Believer

Under the GP surrogate, the unknown function values, $\mathbf{y}_b = \{y_j\}_{j=1}^{q-1}$, at the busy locations, $\mathcal{B} = \{x_j\}_{j=1}^{q-1}$,

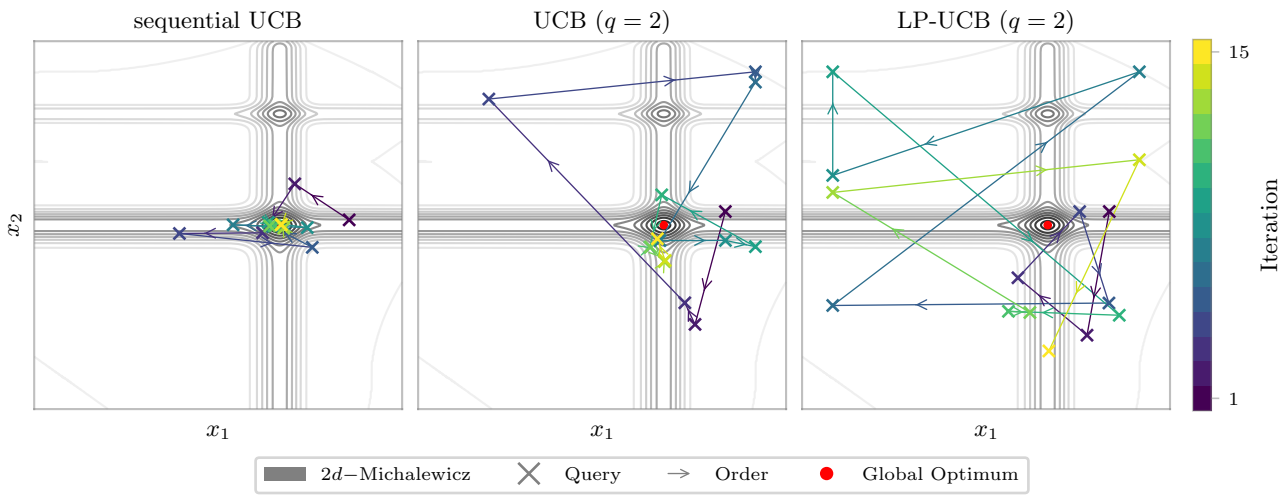

Figure 4: Sample trajectories for the sequential UCB, the asynchronous (standard) UCB, and the purpose-built LP-UCB, with the same initial data. Like sequential UCB, the asynchronous UCB discovers the global optimum (red) within the budget of 15 iterations, despite performing the occasional close query. The *penalization*-based method LP-UCB over-explores the search space and does not find the optimum within the budget. See Figure E.2 for more runs and different methods.

render the acquisition function, $\alpha(x \mid D_n, (\mathcal{B}, \mathbf{y}_b))$, a random function. As Ginsbourger et al. (2011) point out, it is not clear how to deal with this randomness, since the optimization of a random function is not a well-defined task. The proposed solution is to optimize the expected acquisition function, in particular the EI. Due to the intractability of this expectation, this is estimated from *Monte Carlo* samples (see Appendix A.2).

Using a different approach, *penalization*-based methods (LP and LLP) by Alvi et al. (2019) attempt to approximate the expected UCB as

$$\alpha_{LP}(x \mid \mathcal{B}) \approx \mathbb{E}[\alpha_{UCB}(x \mid D_n, D_b) \mid D_n, \mathcal{B}] \tag{6}$$

$$= \int \alpha_{UCB}(x \mid D_n, D_b)\, p(\mathbf{y}_b \mid D_n) d\mathbf{y}_b. \tag{7}$$

Unlike for the EI, we show that for the UCB, this integral is in fact analytically tractable and leads to a well-known heuristic.

Proposition 1 (Expected UCB is the Kriging Believer)**.** *Consider the random Upper Confidence Bound, $\alpha_{UCB}(x|D_n, (\mathcal{B}, \mathbf{y}_b))$, of the GP surrogate posterior, with unobserved function values, $\mathbf{y}_b$, at known busy locations, $\mathcal{B}$. Then it holds that*

$$\begin{aligned}\mathbb{E}[\alpha_{UCB}(x \mid D_n, (\mathcal{B}, \mathbf{y}_b)) \mid D_n, \mathcal{B}] \\ = \alpha_{UCB}(x \mid D_n, (\mathcal{B}, \mu_b))\end{aligned} \tag{8}$$

with μ_b the GP posterior mean at $\mathcal{B}$.

Proof. The proof is given in Appendix C. □

We use the fact that $\mu(\cdot \mid D_n, D_b)$ is linear in $\mathbf{y}_b$ and that $\sigma(\cdot \mid D_n, D_b)$ only depends on the known input locations, X and $\mathcal{B}$, but not on the unknown function values.

From the above result, it can be seen that marginalizing over the values at busy locations gives the UCB we would get by simply assuming the posterior mean, μ_b, at $\mathcal{B}$. This is a *hallucination*-based heuristic well known as the Kriging Believer (Ginsbourger et al., 2010). On the other hand, Proposition 1 then gives an additional insight into the Kriging Believer. We have now shown that, in the case of the UCB, the Kriging Believer is the expected acquisition function.

We have thus shown that the UCB *penalization*-based methods, as well as the Kriging Believer, are constructed from the expected acquisition function, not taking into account higher-order moments of the random acquisition function $\alpha(x \mid D_n, (\mathcal{B}, \mathbf{y}_b))$. This provides a theoretical underpinning for the poor performance of these methods (Alvi et al., 2019; De Ath et al., 2021a), which we further demonstrate in Section 5.1.

4 CONCEPTUAL ARGUMENTS

After presenting some theoretical motivation for why standard acquisition can work well and other heuristics may not, we turn to a more conceptual investigation of Hypothesis 1.

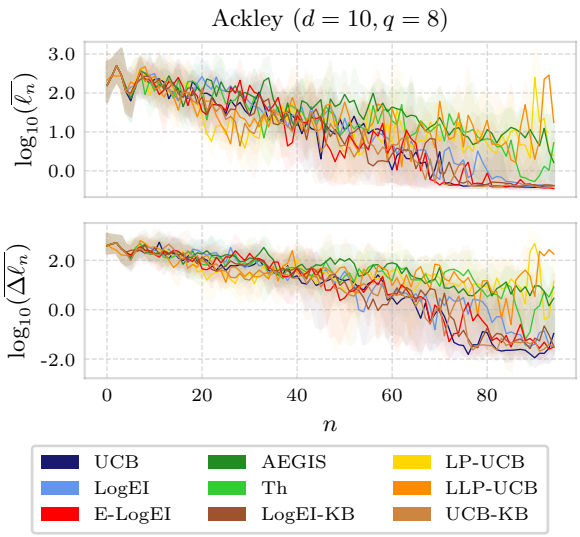

Figure 5: *Top:* The average ARD-RBF kernel lengthscale decreases by about two orders of magnitude from initialization to a plateau for standard methods. Non-standard methods do not all reach this plateau. *Bottom:* The mean absolute difference in subsequent lengthscales drops significantly from initialization to convergence for standard acquisition functions. Several non-standard methods query locations leading to non-converging kernel lengthscales. See Section 4.2 for formal definitions of $\overline{\ell_n}$ and $\overline{\Delta\ell_n}$.

4.1 Each new query is selected under a different—and more informed—GP posterior

As described in line 1 of Algorithm 1, the q workers are initialized at quasi-random locations (e.g., via Halton sequence (Halton, 1964)) in the search space, $\mathcal{X}$, guaranteeing initial diversity. Once a worker finishes its function evaluation, y_{n+1}, at x_{n+1}, the surrogate is updated and a new acquisition is made. This new acquisition is informed by the newly arrived datum (x_{n+1}, y_{n+1}), as well as all data collected by all workers prior, $D_{n+1} = D_n \cup (x_{n+1}, y_{n+1})$, and selected by

$$x' = \arg\max_{x \in \mathcal{X}} \alpha(x \mid D_{n+1}). \tag{9}$$

Observation 1 (Hypothesis 1 neglects GP surrogate update)**.** *Hypothesis 1 states that the new input location, x', will either be redundant or even repeated. But this fails to account for the update to the surrogate likelihood via the new datum, (x_{n+1}, y_{n+1}), as well as to the surrogate kernel hyperparameters, ϕ. In fact, existing works presented in Appendix A.2 never discuss the intermediate update of the surrogate at all.*

Building intuition The stated goal of introducing purpose-built acquisition functions for asynchronous BO was to prevent redundant queries. Observation 1 suggests that the problem of querying the same, or very similar, locations in asynchronous BO does not trivially arise from using standard acquisition functions. Showing one asynchronous BO step for the UCB, the locally *penalized* UCB (LP-UCB) and the *hallucinating* Kriging Beliver UCB, Figure 3 illustrates that the intermediate update can, at least in principle, be effective in guiding the subsequent acquisition. Without a need to explicitly enforce diversity, the new datum shifts the optimum of the UCB into a promising region, far from previous and currently busy queries. Importantly, the purpose-built methods taking into account the busy locations in rather elaborate ways end up querying the same point as the standard UCB. This effect persists in the following iterations, as can be seen in Figure E.1.

Moreover, from Observation 1, methods explicitly enforcing diversity in queries could be expected to suffer from an inefficient over-exploration of the search space. This is because, by preventing exploitation, the algorithm is forced to query less promising areas, which should have already been eliminated from consideration. We illustrate this in Figure 4, which shows the sample trajectories for three variants of the UCB heuristic. The optimally informed sequential UCB recovers the optimum within few iterations. An over-exploration of the search space by the *penalization*-based method LP-UCB (Alvi et al., 2019) prevents it from reaching the global optimum within the iteration budget. While the asynchronous UCB occasionally queries close to previous locations, against which there is no theoretical guarantee, it offers a better exploration-exploitation trade-off and thus reaches close to the global optimum.

4.2 The kernel and likelihood updates naturally facilitate a good exploration-exploitation trade-off

Good results in BO require good exploration-exploitation trade-offs Srinivas et al. (2010). The concern of Hypothesis 1 is that a new observation will not produce a sufficiently large update to encourage exploration. We now disentangle the respective contribution to the exploration-exploitation trade-off of the two distinct updates presented in Observation 1. To this end, we define

$$\overline{\ell_n} \equiv \frac{1}{d} \sum_{j=1}^{d} \ell_{n,j} \tag{10}$$

and

$$\overline{\Delta\ell_n} \equiv \frac{1}{d} \sum_{j=1}^{d} |\ell_{n,j} - \ell_{n-1,j}|, \tag{11}$$

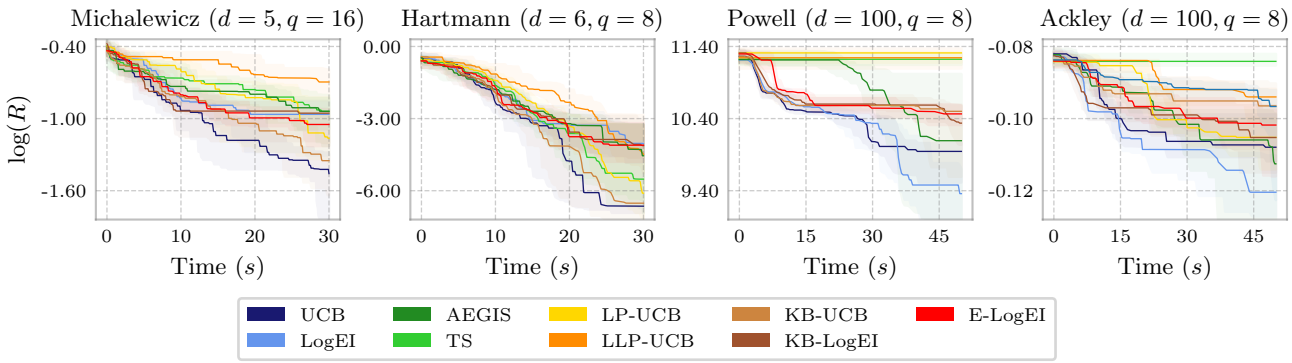

Figure 6: **Standard acquisition dominates on synthetic tasks.** This observation is robust to the dimensionality and the number of workers, whereas the performance of the purpose-built methods is always worse and more sensitive to task parameters (see also Figures G.1 and G.2).

where $\ell_{n,j} > 0$ is the lengthscale corresponding to dimension $j \in [d]$ of an ARD kernel, $k_\phi(\cdot, \cdot)$, as introduced in Section 2, at iteration n of Algorithm 1. While $\overline{\ell_n}$ is a measure for the covariance between function values under the GP-surrogate averaged over all d axes, $\overline{\Delta\ell_n}$ reflects the change in this covariance compared to the previous iteration.

Exploration phase and kernel update The updated kernel lengthscales, $\ell_{n+1} \in \mathbb{R}^d$, in particular, will affect the shape of the surrogate posterior and ultimately the acquisition. The lengthscales are updated together with the other kernel hyperparameters from ϕ_n to ϕ_{n+1} to fit the updated data, D_{n+1} (line 11 in Algorithm 1). This is done by optimizing the unnormalized posterior, $p(\phi|\mathbf{y}_{n+1})$, as

$$\phi_{n+1} = \arg\max_{\phi} \ \log p(\mathbf{y}_{n+1} \mid \phi) + \log p(\phi), \qquad (12)$$

in a gradient-based manner.

In Figure 5, we find empirically that the magnitude of this update is largest in the early stages of the optimization and essentially vanishes towards the end. Intuitively, in the exploration phase with little data, the new datum will carry significant information on the lengthscales. Being a global property of the surrogate, the lengthscales can shift the acquisition function optimum, even if the new datum is in an unpromising region, thereby facilitating exploration of the search space.

Exploitation phase and likelihood update For most common kernels, the nature of the likelihood update is local, forcing the GP posterior mean, $\mu(\cdot|D_{n+1})$, to pass close by the new datum. That is, $y_{n+1} \approx \mu(x_{n+1}|D_{n+1})$, depending on the observation noise level, η^2. Also, the uncertainty, expressed via the posterior variance $\sigma^2(\cdot|D_{n+1})$, collapses to at most the observation noise, η^2, at the input x_{n+1}, reflecting the gain in information at and around this location. Note that this collapse occurs even if y_{n+1} already lies on the posterior mean, i.e., $y_{n+1} = \mu(x_{n+1} \mid D_n)$. This local update changes the acquisition function around (x_{n+1}, y_{n+1}).

In the exploitation phase, kernel lengthscales have effectively converged for standard acquisition methods (Figure 5). Thus, we can no longer rely on the kernel update later in the optimization, but we argue that this may not even be necessary. In the exploitation phase, queries *should* lie close together, and the acquisition function optimum only needs to be shifted locally in each update. As argued above, this local shift is exactly what the likelihood update can provide.

5 EXPERIMENTAL EVALUATION

We support our theoretical and conceptual insights from Sections 3 and 4 with experiments on a range of synthetic, real-world and hyperparameter tuning tasks. After demonstrating excellent performance of standard acquisition functions (Section 5.1), we investigate the exploration-exploitation trade-off made by both standard and purpose-built acquisition functions (Section 5.2).

5.1 Optimization tasks

Following common practice, we report the logarithm of the simple regret, R, $\log(R) = \log|f^* - y_n^*|$, where f^* is the known function optimum. We plot the median of the $\log(R)$ together with the inter-quartile range. For details on the implementation[1] and tasks, please refer to Appendices D and F.

[1]We make our code available at https://github.com/fortuinlab/SAIS-BO

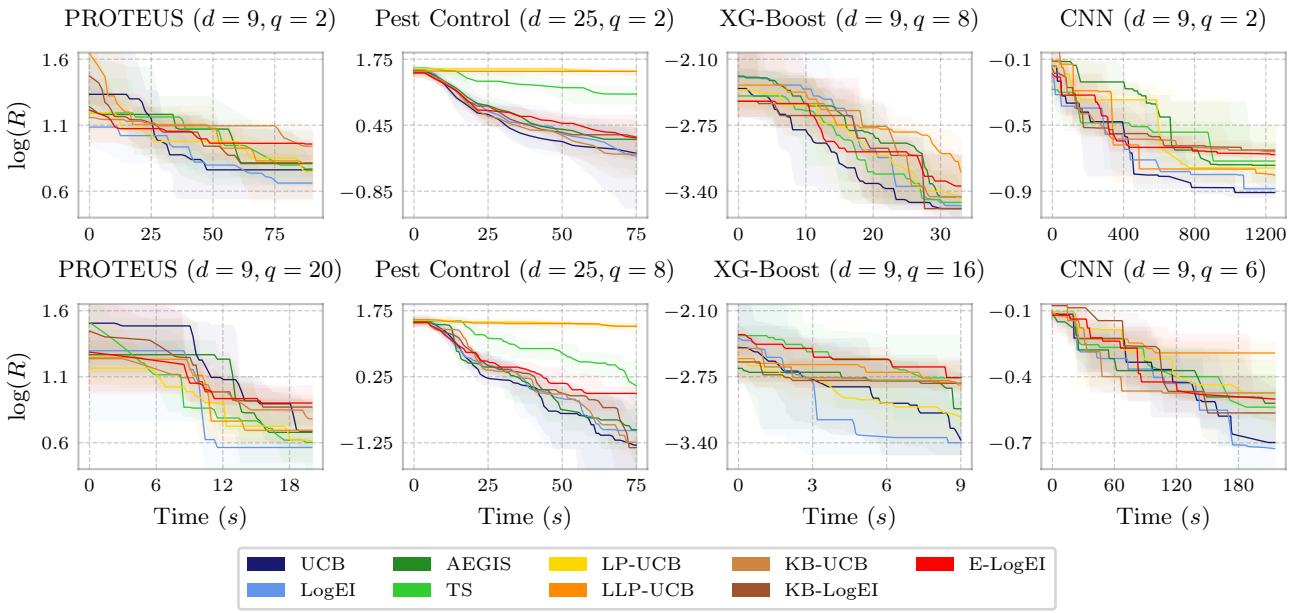

Figure 7: **Standard acquisition performs well on real-world and hyperparameter tuning tasks and benefits more from an increase in the number of workers than purpose built methods.** Both rows show the same optimization tasks, but with different numbers of workers, q. From the axes it can be seen how an increase in evaluation resources speeds up the optimization. Importantly, standard acquisition often benefits disproportionately from more workers, compared to existing methods.

Synthetic test functions The acquisition rules are compared on synthetic test functions across a range of dimensions and numbers of workers. All experiments are repeated 20 times with different initializations. In order to simulate the asynchronous setting, the evaluation time is sampled from a half-normal distribution with scale parameter $\theta = \sqrt{\pi/2}$ (Alvi et al., 2019; De Ath et al., 2021a). We choose functions used in works outlined in Appendix A.2, to enable comparison of results.

In Figure 6, it can be seen that no purpose-built method outperforms both the standard UCB and LogEI. While the KB-UCB does almost as well as the UCB on the $5d$–Michalewicz and $6d$–Hartmann functions, the LogEI is superior on the very high–dimensional Powell and Ackley functions. The *penalization*-based methods, as well as TS, are unable to make meaningful progress in this vast search space. A large number of additional results on more test functions and different configurations (d, q), and with larger time budget can be found in Appendix G.

Real-world tasks The PROTEUS and Pest Control tasks showcase the effectiveness on problems beyond standard synthetic functions. In PROTEUS (Lichtenberg et al., 2021), a physics-based simulator for the evolution of planets, the objective is to retrieve the initial condition given a simulated planet, while Pest Control requires making a decision on herbicide at five different stations. See Appendix F.2 for a more detailed description.

Importantly, the PROTEUS experiment demonstrates how our findings continue to hold true, even in the very high worker setting of $q = 20$ (see Figure 7). This is difficult to reconcile with Hypothesis 1, where one might expect query redundancy to be exacerbated by a large number of workers. But, Figure 7 reveals that this is not the case for either of these real-world experiments.

Hyperparameter tuning tasks To demonstrate the robustness of our findings, experiments are performed on two hyperparameter tuning tasks. Due to the large computational cost of these experiments (which is precisely what makes them relevant in this work on BO), they are repeated with 9 different initializations. The tasks are briefly outlined below, and the results are shown in Figure 7. In Appendix F.2 we discuss the hyperparameter tuning tasks in more detail.

For XG-Boost, we consider nine hyperparameters, which we tune to optimize the 5-fold cross-validation accuracy. This score is computed from the classification performance on the UCI Breast Cancer data (Wolberg et al., 1993). We also learn the hyperparameters of a 6-layer CNN pipeline to optimize validation accuracy, after 20 epochs of training on CIFAR10 (Krizhevsky

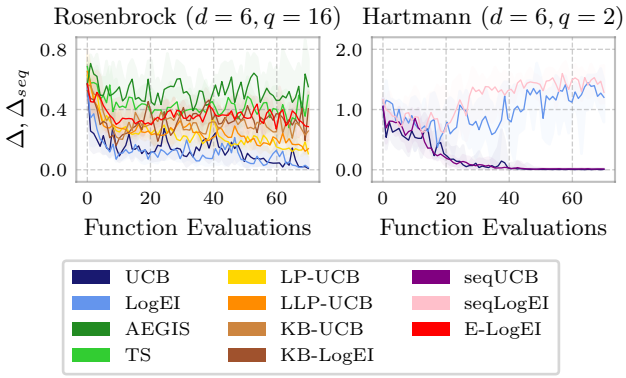

Figure 8: **Standard acquisition does not systematically repeat queries and performs an exploration-exploitation trade-off similar to that of its optimally informed sequential counterpart.** It can be seen that standard acquisition methods do not repeat queries, and that standard acquisitions query at similar distances to previous queries in the asynchronous setting as in the sequential setting.

and Hinton, 2009).

Contrary to Hypothesis 1, we see in Figure 7 that standard methods are able to effectively utilize parallel workers to accelerate the optimization, without the need for explicit diversity. The LogEI does extremely well on both tasks, while the UCB effectively matches this performance on the CNN task. A key insight here is that standard acquisition functions benefit disproportionately from an increase in the number of workers, q, compared to existing methods.

5.2 Analysis of distances of queries from busy locations

We now turn to an empirical investigation of our conceptual analysis in Section 4.1. Hypothesis 1 claims that in the absence of regularization, query distances to busy locations will be small or even zero. To quantify this behavior, we consider the distances of asynchronous queries to currently busy locations. As a gold standard, we also introduce the distances of the optimally informed sequential BO queries to the $q-1$ most recently sampled points. This allows for comparison of the exploration-exploitation trade-off made by all methods considered. Formally, we define the distance, Δ, of the asynchronous query, x', from busy locations, $\mathcal{B}$, as

$$\Delta \equiv \min_{x_j \in \mathcal{B}} \|x' - x_j\|_2, \tag{13}$$

and analogously the distance of the nth query in sequential BO to the last $q-1$ sequential queries as

$$\Delta_{seq} \equiv \min_{i \in [q-1]} \|x_n - x_{n-i}\|_2. \tag{14}$$

In Figure 8, we show the median and inter-quartile range of these distances for a number of test functions, across 20 independent runs. It can be seen that existing methods query further from busy locations than standard acquisition, as is intended. But, contrary to Hypothesis 1, standard acquisition does not systematically perform repeated queries from the very start of the optimization. In fact, the standard UCB and, in some cases, also LogEI, exhibit the desirable transition from an initial exploratory phase (Δ large) to an exploitation stage (Δ small).

Perhaps even more surprising is the comparison of standard acquisition functions to their respective sequential counterparts. Figure 8 shows that standard acquisition in the asynchronous setting very closely matches the query distances of the sequential one. Interestingly, this analysis reveals that the lack of an exploitation phase of LogEI on, e.g., Hartmann is not due to issues described in Hypothesis 1, but simply a characteristic of the LogEI itself.

This analysis further supports the conceptual arguments made in Section 4. Existing purpose-built methods enforcing diversity seem to suffer from over-exploration, subsequently leading to a weaker exploitation phase of the optimization. On the other hand, standard acquisition performs an exploration-exploitation trade-off closely aligned with that of the optimally informed sequential BO.

6 CONCLUSION

A recent line of research falls under the overarching theme: *BO is easier than previously thought and simple approaches work surprisingly well* (Hvarfner et al., 2024; Xu et al., 2025; Ament et al., 2023). In this work, we add to this by demonstrating that simple and standard acquisition functions perform well in asynchronous BO, while existing methods for asynchronous BO are founded on the unsupported assumption in Hypothesis 1.

While the literature has not presented a purpose-built method superior to standard acquisition in asynchronous BO, this does not mean such a method cannot exist. However, we have shown that existing methods in fact often degrade performance by enforcing query-diversity, likely by inducing an inefficient over-exploration of the search space.

While further exploration may often be needed to find the global optimum, it is currently unclear how to conduct this efficiently. In future research, we plan to further explore mechanisms taking into account the busy locations to achieve efficient exploration in the asynchronous BO algorithm.

Acknowledgements

We thank Tommy Rochussen for helpful discussions. VF was supported by the Branco Weiss Fellowship.

References

Ahsan S. Alvi, Binxin Ru, Jan Calliess, Stephen J. Roberts, and Michael A. Osborne. Asynchronous batch bayesian optimisation with improved local penalisation. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

Sebastian Ament, Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Unexpected improvements to expected improvement for Bayesian optimization. *Advances in Neural Information Processing Systems*, 36:20577–20612, 2023.

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G Wilson, and Eytan Bakshy. BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. *Advances in neural information processing systems*, 33:21524–21538, 2020.

Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing*, 16(5):1190–1208, 1995.

Yutian Chen, Aja Huang, Ziyu Wang, Ioannis Antonoglou, Julian Schrittwieser, David Silver, and Nando de Freitas. Bayesian optimization in AlphaGo. *arXiv preprint arXiv:1812.06855*, 2018.

Tristan Cinquin, Stanley Lo, Felix Strieth-Kalthoff, Alan Aspuru-Guzik, Geoff Pleiss, Robert Bamler, Tim GJ Rudner, Vincent Fortuin, and Agustinus Kristiadi. What actually matters for materials discovery: Pitfalls and recommendations in Bayesian optimization. In *AI for Accelerated Materials Design-ICLR 2025*, 2025.

George De Ath, Richard M Everson, and Jonathan E Fieldsend. Asynchronous ε-greedy Bayesian optimisation. In *Uncertainty in Artificial Intelligence*, pages 578–588. PMLR, 2021a.

George De Ath, Richard M Everson, Alma AM Rahat, and Jonathan E Fieldsend. Greed is good: Exploration and exploitation trade-offs in Bayesian optimisation. *ACM Transactions on Evolutionary Learning and Optimization*, 1(1):1–22, 2021b.

Thomas Desautels, Andreas Krause, and Joel W. Burdick. Parallelizing exploration–exploitation tradeoffs in Gaussian Process bandit optimization. *Journal of Machine Learning Research*, 15(1):3873–3923, 2014.

Romain Egelé, Isabelle Guyon, Venkatram Vishwanath, and Prasanna Balaprakash. Asynchronous decentralized Bayesian optimization for large scale hyperparameter optimization. In *2023 IEEE 19th International Conference on e-Science (e-Science)*, pages 1–10. IEEE, 2023.

Andreas Ekström, Christian Forssén, Christos Dimitrakakis, D Dubhashi, Håkan T Johansson, Azam Sheikh Muhammad, Hans Salomonsson, and Alexander Schliep. Bayesian optimization in ab initio nuclear physics. *Journal of Physics G: Nuclear and Particle Physics*, 46(9):095101, 2019.

Trevor S Frisby, Zhiyun Gong, and Christopher James Langmead. Asynchronous parallel Bayesian optimization for AI-driven cloud laboratories. *Bioinformatics*, 37(Supplement_1):i451–i459, 2021.

Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. *Advances in neural information processing systems*, 31, 2018.

David Ginsbourger, Rodolphe Le Riche, and Laurent Carraro. A multi-points criterion for deterministic parallel global optimization based on Kriging. In *NCP07*, 2007.

David Ginsbourger, Rodolphe Le Riche, and Laurent Carraro. Kriging is well-suited to parallelize optimization. In *Computational intelligence in expensive optimization problems*, pages 131–162. Springer, 2010.

David Ginsbourger, Janis Janusevskis, and Rodolphe Le Riche. *Dealing with asynchronicity in parallel Gaussian process based global optimization.* PhD thesis, Mines Saint-Etienne, 2011.

Javier González, Zhenwen Dai, Philipp Hennig, and Neil Lawrence. Batch Bayesian optimization via local penalization. In *Artificial intelligence and statistics*, pages 648–657. PMLR, 2016.

Ryan-Rhys Griffiths and José Miguel Hernández-Lobato. Constrained Bayesian optimization for automatic chemical design using variational autoencoders. *Chemical science*, 11(2):577–586, 2020.

John H Halton. Algorithm 247: Radical-inverse quasi-random point sequence. *Communications of the ACM*, 7(12):701–702, 1964.

Carl Hvarfner, Erik Orm Hellsten, and Luigi Nardi. Vanilla bayesian optimization performs great in high dimensions. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Janis Janusevskis, Rodolphe Le Riche, David Ginsbourger, and Ramunas Girdziusas. Expected improvements for the asynchronous parallel global opti-

mization of expensive functions: Potentials and challenges. In *International Conference on Learning and Intelligent Optimization*, pages 413–418. Springer, 2012.

Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13:455–492, 1998.

Kirthevasan Kandasamy, Akshay Krishnamurthy, Jeff Schneider, and Barnabás Póczos. Parallelised Bayesian optimisation via Thompson sampling. In *International conference on artificial intelligence and statistics*, pages 133–142. PMLR, 2018.

Aaron Klein, Stefan Falkner, Simon Bartels, Philipp Hennig, and Frank Hutter. Fast Bayesian optimization of machine learning hyperparameters on large datasets. In *Artificial intelligence and statistics*, pages 528–536. PMLR, 2017.

Yuki Koyama and Masataka Goto. Bo as assistant: Using Bayesian optimization for asynchronously generating design suggestions. In *Proceedings of the 35th annual ACM symposium on user interface software and technology*, pages 1–14, 2022.

Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in Gaussian Processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(2):235–284, 2008.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Tim Lichtenberg, Dan J. Bower, Mark Hammond, Ryan Boukrouche, Patrick Sanan, Shang-Min Tsai, and Raymond T. Pierrehumbert. Vertically Resolved Magma Ocean-Protoatmosphere Evolution: H_2, H_2O, CO_2, CH_4, CO, O_2, and N_2 as Primary Absorbers. *Journal of Geophysical Research (Planets)*, 126(2):e06711, February 2021. doi: 10.1029/2020JE006711.

Jonas Močkus. On Bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974 6*, pages 400–404. Springer, 1975.

Harrison Nicholls, Tim Lichtenberg, Dan J. Bower, and Raymond Pierrehumbert. Magma Ocean Evolution at Arbitrary Redox State. *Journal of Geophysical Research (Planets)*, 129(12):2024JE008576, December 2024. doi: 10.1029/2024JE008576.

Changyong Oh, Jakub Tomczak, Efstratios Gavves, and Max Welling. Combinatorial Bayesian optimization using the graph Cartesian product. *Advances in Neural Information Processing Systems*, 32, 2019.

Daniel Packwood et al. *Bayesian optimization for materials science*, volume 3. Springer, 2017.

Ryan Roussel, Auralee L Edelen, Tobias Boltz, Dylan Kennedy, Zhe Zhang, Fuhao Ji, Xiaobiao Huang, Daniel Ratner, Andrea Santamaria Garcia, Chenran Xu, et al. Bayesian optimization algorithms for accelerator physics. *Physical review accelerators and beams*, 27(8):084801, 2024.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, 2010.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Anh Tran, Mike Eldred, Tim Wildey, Scott McCann, Jing Sun, and Robert J Visintainer. aphBO-2GP-3B: A budgeted asynchronous parallel multi-acquisition for constrained Bayesian optimization on high-performing computing architecture. *Structural and Multidisciplinary Optimization*, 65(4):132, 2022.

Ke Wang and Alexander W Dowling. Bayesian optimization for chemical products and functional materials. *Current Opinion in Chemical Engineering*, 36: 100728, 2022.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

James Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Deisenroth. Efficiently sampling functions from Gaussian process posteriors. In *International Conference on Machine Learning*, pages 10292–10302. PMLR, 2020.

William Wolberg, Olvi Mangasarian, Nick Street, and W. Street. Breast cancer Wisconsin (diagnostic). UCI Machine Learning Repository, 1993. https://doi.org/10.24432/C5DW2B.

Jia Wu, Xiu-Yun Chen, Hao Zhang, Li-Dong Xiong, Hang Lei, and Si-Hao Deng. Hyperparameter optimization for machine learning models based on Bayesian optimization. *Journal of Electronic Science and Technology*, 17(1):26–40, 2019.

Zhitong Xu, Haitao Wang, Jeff M Phillips, and Shandian Zhe. Standard Gaussian process is all you need for high-dimensional Bayesian optimization. In *The Thirteenth International Conference on Learning Representations*, 2025.

Shuhan Zhang, Fan Yang, Dian Zhou, and Xuan Zeng. An efficient asynchronous batch Bayesian optimization approach for analog circuit synthesis. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2020.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. Yes. See Algorithm 1 and Appendix A.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. Yes. See Section 3.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. Yes. https://github.com/fortuinlab/SAIS-BO

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. Yes. See Appendix B.
   (b) Complete proofs of all theoretical results. Yes. See Appendix B.
   (c) Clear explanations of any assumptions. Yes. See Appendix B.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). Yes. https://github.com/fortuinlab/SAIS-BO
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). Yes. See Appendix D.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). Yes. See Section 5.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). Yes. See Appendix H.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. Yes. See Appendices D and F.
   (b) The license information of the assets, if applicable. Yes. See Appendices D and F and https://github.com/fortuinlab/SAIS-BO.
   (c) New assets either in the supplemental material or as a URL, if applicable. Yes. https://github.com/fortuinlab/SAIS-BO
   (d) Information about consent from data providers/curators. Not Applicable.
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. Not Applicable.

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. Not Applicable.
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. Not Applicable.
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. Not Applicable.

A Acquisition function details

The probabilistic nature of the surrogate model allows the acquisition function to reason about uncertainty in $f(\cdot)$. This, in turn, guides the tradeoff of exploration and exploitation on the search space $\mathcal{X}$. Formally, the next query location, x', is proposed by an acquisition rule, with acquisition function $\alpha : \mathcal{X} \mapsto \mathbb{R}$, as

$$x' = \arg\max_{x \in \mathcal{X}} \ \alpha(x \mid D_n). \tag{15}$$

We drop the conditioning on D_n for notational convenience, unless explicitly mentioning it is necessary. This section presents a number of standard and asynchronous acquisition rules relevant to our work.

A.1 Standard acquisition

Upper Confidence Bound The GP Upper Confidence Bound (UCB) (Srinivas et al., 2010) is a simple heuristic based on optimizing a quantile of the credibility interval, for example, the 95% outcome. It is defined as

$$\alpha_{UCB}(x) = \mu(x \mid D_n) + \sqrt{\beta}\sigma(x \mid D_n), \tag{16}$$

with $\mu(\cdot)$ and $\sigma(\cdot)$ as in Equations (2) and (3).

It can be seen as a weighted sum of the posterior mean and standard deviation, where the relative contribution of each summand is set via the hyperparameter β. The exploration-exploitation tradeoff is controlled by β, which may be set to a fixed value or according to some schedule (Srinivas et al., 2010). In our work we keep $\beta = 2$ fixed.

Expected Improvement The Expected Improvement (EI) (Močkus, 1975; Jones et al., 1998) acquisition function assigns utility to an input location, $x \in \mathcal{X}$, according to how much the associated function value, $f(x)$, is expected (under the surrogate posterior) to improve on the best function value, $y_n^* = \max_{i \in [n]} \mathbf{y}$, observed so far. Formally,

$$\alpha_{EI}(x) = \mathbb{E}_{f \mid D_n}[\max(f(x) - y_n^*, 0)]. \tag{17}$$

For numerical stability in the optimization, this work uses the natural logarithm of EI (Ament et al., 2023).

Random Search While there are many ways to perform BO with random search acquisition, the simplest variant sampling

$$x' \sim \mathcal{U}[0, 1]^d \tag{18}$$

is used in this work. A more elaborate scheme would, e.g., be to sample from a low-discrepancy quasi-random sequence, such as the Halton sequence (Halton, 1964).

A.2 Asynchronous acquisition rules

Monte Carlo sampling Ginsbourger et al. (2011) present a *Monte Carlo* sampling-based estimate of the expected EI, where the expectation is with respect to the unknown function values, $\mathbf{y}_b$, under the GP surrogate. In particular, they form

$$\alpha_{E-EI}(x \mid \mathcal{B}) = \frac{1}{N} \sum_{i=1}^{N} \alpha_{EI}(x \mid D_n, (\mathcal{B}, \mathbf{y}_{b,i})), \tag{19}$$

where $\{\mathbf{y}_{b,i}\}_{i=1}^N$ are i.i.d. samples from the GP surrogate posterior predictive

$$p(\mathbf{y}_b \mid D_n) = \mathcal{N}(\mathbf{y}_b \mid \mu_b, \Sigma_b), \tag{20}$$

with

$$\mu_b = K_{\mathcal{B}X}(K_{XX} + \eta^2 \mathbf{I}_n)^{-1}\mathbf{y} \in \mathbb{R}^{q-1} \tag{21}$$

$$\Sigma_b = K_{\mathcal{B}\mathcal{B}} - K_{\mathcal{B}X}(K_{XX} + \eta^2 \mathbf{I}_n)^{-1} K_{X\mathcal{B}} \in \mathbb{R}^{(q-1)\times(q-1)}. \tag{22}$$

While Ginsbourger et al. (2011) do this for the EI, this approach may be taken for any analytical acquisition function.

Hallucination A further approach to dealing with the unobserved function values is *hallucination*-based methods, such as the Kriging Believer (Ginsbourger et al., 2010). Here, the unknown values, $\mathbf{y}_b$, are simply replaced by their posterior means under the GP surrogate when forming the acquisition function. In particular, for any analytic acquisition function, $\alpha(\cdot)$,

$$\alpha_{KB}(x \mid \mathcal{B}) = \alpha(x \mid D_n, (\mathcal{B}, \mu_b)), \tag{23}$$

with μ_b as in Equation (21).

While developed for synchronous q-batch construction, the constant liar heuristic offers an additional mechanism to account for locations under evaluation (Ginsbourger et al., 2007).

Thompson sampling Thompson sampling (TS) is an acquisition rule based on sampling the surrogate posterior (Thompson, 1933). The goal of the approach is to sample points in the input space where the maximum is most likely to be. Consider a function $g \sim p(\cdot \mid D_n)$ which is sampled from the surrogate posterior. The probability that a particular input is the maximizer, x^*, is given by

$$p(x^*|D_n) = \int p(x^* \mid g)\ p(g \mid D_n) dg \tag{24}$$

$$= \int \delta_{\arg\max_{x \in \mathcal{X}} g(x)}(x^*)\ p(g \mid D_n) dg, \tag{25}$$

we can sample from $p(x^*|D_n)$ in a simple two step procedure. First, draw $g \sim p(\cdot \mid D_n)$, and then return $x' = \arg\max_{x \in \mathcal{X}}\ g(x)$.

In this work, all Thompson samples are drawn using the state-of-the-art decoupled sampling method by Wilson et al. (2020), giving differentiable posterior function samples. This significantly enhances performance over the standard approach of sampling function values at a discrete set of inputs.

In the context of asynchronous batch BO, this method was proposed, analyzed theoretically, and evaluated empirically by Kandasamy et al. (2018). In line with Hypothesis 1, they motivate the need for their method with the occurrence of redundant function evaluations in standard asynchronous BO, while acknowledging that queries will not be exactly repeated. In the one real-world experiment performed by Kandasamy et al. (2018), this method did not outperform standard EI. Moreover, this work does not use the numerically advantageous LogEI, since it precedes the work by Ament et al. (2023). This likely explains the inferior performance of standard EI in their work compared to ours.

AEGIS Proposed by De Ath et al. (2021a), Asynchronous ϵ-Greedy Global Search (AEGIS) is an acquisition rule probabilistically combining three heuristics: *(i)* performs TS with probability ϵ_T, *(ii)* chooses randomly from the Pareto Frontier for the two objectives of maximizing the mean ($\mu(\cdot)$) and the variance ($\sigma^2(\cdot)$) with probability ϵ_P, and *(iii)* otherwise optimizes the surrogate mean. The probabilities for the different modes are set as

$$\epsilon_T = \epsilon_P = \epsilon/2 \tag{26}$$

$$\epsilon = \min\{2/\sqrt{d}, 1\}, \tag{27}$$

such that the tendency to exploit the surrogate mean decays as $1/\sqrt{d}$.

De Ath et al. (2021a) are motivated by Hypothesis 1 and the shortcomings of TS. They choose not to compare their method to standard acquisition rules like EI or UCB.

(Local) Lipschitz penalization The asynchronous Lipschitz penalization (LP) method, called "PLAyBOOK" by Alvi et al. (2019), aims to ensure acquisition diversity by creating exclusion cones in the acquisition surface, centered on the busy locations, $\mathcal{B}$. This is based on work for sequentially constructing a q-batch, in synchronous batch BO (González et al., 2016) (Figure 2). The extent of the penalization depends on an estimate of the objective function's Lipschitz constant, L, and the global optimum, f^*. Formally, Alvi et al. (2019) design the local penalizer centered at busy location x_j, $\varphi : \mathcal{X} \mapsto [0, 1]$, as

$$\varphi(x \mid x_j) = \min\left\{\frac{\hat{L}\|x - x_j\|}{\mid \mu(x_j) - y_n^* \mid + \gamma\sigma(x_j)}, 1\right\}. \tag{28}$$

The objective optimum is estimated as the best function value found so far, $y_n^* = \max_{i \in [n]} \mathbf{y}$. The Lipschitz constant is estimated from the gradient of the posterior mean as $\hat{L} = \max_{x \in \mathcal{X}} \nabla \mu(x)$. Any analytic acquisition function, $\alpha(\cdot)$, may then be locally penalized at a set of busy locations $\mathcal{B} = \{x_j\}_{j=1}^{q-1}$ as

$$\alpha_{LP}(x \mid \mathcal{B}) = \alpha_{UCB}(x) \prod_{j=1}^{q-1} \varphi(x \mid x_j). \tag{29}$$

While this is presented as an approximation,

$$\alpha_{LP}(x \mid \mathcal{B}) \approx \mathbb{E}[\alpha_{UCB}(x \mid D_n, D_b) \mid D_n, \mathcal{B}], \tag{30}$$

we show in Proposition 1 that in fact

$$\mathbb{E}[\alpha_{UCB}(x \mid D_n, D_b) \mid D_n, \mathcal{B}] = \alpha_{KB}(x \mid \mathcal{B}). \tag{31}$$

Judging from Figures 3 and E.1, the quality of this approximation, $\alpha_{LP}(\cdot \mid \mathcal{B}) \approx \alpha_{KB}(\cdot \mid \mathcal{B})$, is questionable, albeit both $\alpha_{LP}(\cdot \mid \mathcal{B})$ and $\alpha_{KB}(\cdot \mid \mathcal{B})$ seem to have a common optimum.

Additionally, Alvi et al. (2019) present a version of this where the Lipschitz constant is not shared by all $q-1$ penalizers, but estimated locally around the respective busy location. The search spaces $\{\mathcal{X}_j\}_{j=1}^{q-1}$ for the local Lipschitz (LLP) estimation are then defined through the kernel lengthscales. In particular, $\mathcal{X}_j \subset \mathcal{X}$ is a hyper rectangle centered on x_j, with side lengths equal to the lengthscales of the respective dimensions.

Following Hypothesis 1, they motivate their method with the danger of repeated and redundant queries at and in the vicinity of busy locations, $\mathcal{B}$. In their work, they do not compare their method to standard acquisition rules like EI or UCB.

From Figure 5 it can be seen that these methods (LP-UCB and LLP-UCB) learn relatively long lengthscales. This can be expected to result in rather small estimated Lipschitz constants, $\hat{L}$, which in turn results in penalization of large regions around the busy locations. This is then likely the driver of the over-exploration observed for these methods (see, e.g., Figure 4).

B Formal statement and proof of Theorem 1

Setting and notation Let $\mathcal{X} \subset [0,1]^d$ be compact and the objective $f \sim \mathcal{GP}(0, k_\phi)$ with $k_\phi(x,x) \leq 1$. Given a finite subset $A = \{x_1, x_2, ..., x_n\} \subset \mathcal{X}$, we have $(f_A)_i = f(x_i)$ and $(\epsilon_A)_i \sim \mathcal{N}(0, \eta^2)$. Observations are $y_A = f_A + \epsilon_A \in \mathbb{R}^n$. In the following, the index j refers to the jth acquisition being made by Algorithm 1. The set of *completed* observations (x, y) available at step j is denoted by $\mathcal{D}_j$.[2] In line with this, $\mu_{\mathcal{D}_j}$ and $\sigma_{\mathcal{D}_j}$ denote the GP posterior mean and standard deviation constructed from data $\mathcal{D}_j$. At step j, with completed set $\mathcal{D}_j$, our policy asynchronous UCB selects

$$x_j = \arg\max_{x \in \mathcal{X}} \; U_j(x), \tag{32}$$

with

$$U_j(x) = \mu_{\mathcal{D}_j}(x) + \sqrt{\beta_j}\, \sigma_{\mathcal{D}_j}(x), \tag{33}$$

for a non-decreasing sequence β_j.

Relevant quantities We introduce the quantity ξ_q, which was studied by Krause et al. (2008); Desautels et al. (2014) and also used by Kandasamy et al. (2018). Informally, it places an upper bound on the information about f in the $q-1$ pending evaluations at busy locations $\mathcal{B}$. Formally we have

$$\max_{\mathcal{B} \subset \mathcal{X}, |\mathcal{B}| < q} I\left(f; y_{\mathcal{B}} \mid \mathcal{D}_n\right) \leq \frac{1}{2} \log\left(\xi_q\right), \tag{34}$$

[2] Note the difference to D_n, a set of n completed evaluations (cf. Section 2).

with I the Shannon mutual information. Further, we make use of the maximum information gain (MIG) after n evaluations, defined as

$$\Psi_n = \max_{A \subset \mathcal{X}, |A|=n} I\left(f; y_A\right). \tag{35}$$

A sequence β_n satisfying the assumptions on which the relevant Lemmas rest is

$$\beta_j = 4(d+1)\log(j) + 2d\log(dab\sqrt{\pi}), \tag{36}$$

with d the problem dimensionality and constants a and b from Assumption 8 in Kandasamy et al. (2018). Note that our choice of keeping $\beta_j = 2$ fixed does not satisfy the conditions required for the theorem to hold, but seems to work well nonetheless (see Appendix A.1).

Next, we state a formal version of Theorem 1.

Theorem 2 (Bound on BSR(n) for asynchronous UCB)**.** *Let $f \sim \mathcal{GP}(0, k_\phi)$, where $k_\phi : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ satisfies Assumption 8 in Kandasamy et al. (2018) and, w.l.o.g., $k_\phi(x,x) \leq 1$. Then, for the UCB acquisition function used in Algorithm 1, the Bayes simple regret after n queries can be bounded as*

$$\mathrm{BSR}(n) \leq \frac{\pi^2 + \sqrt{2\pi}}{12n} + \sqrt{\frac{2\xi_q \beta_n \Psi_n}{n\log(1+\eta^{-2})}}. \tag{37}$$

Proof. Our proof heavily relies on Kandasamy et al. (2018) and we will refer to relevant results of this work throughout. For a given number of steps, n, we seek a bound on the Bayes simple regret (BSR) defined as

$$\mathrm{BSR}(n) = \mathbb{E}[f(x^*) - \max_{j \in [n]} f(x_j)], \tag{38}$$

where the expectation is w.r.t. the prior on f, the observation noises $\{\epsilon_j\}_{j=1}^n$.

We first define and deconstruct the Bayes cumulative regret (BCR, Kandasamy et al. (2018)) as follows.

$$\begin{aligned}
\mathrm{BCR}(n) &= \sum_{j=1}^n \mathbb{E}[f(x^*) - f(x_j)] && (39)\\
&= \sum_{j=1}^n \mathbb{E}[f(x^*) - f([x^*]_j) + f([x^*]_j) - U_j([x^*]_j) + U_j([x^*]_j) - f(x_j)] && (40)\\
&= \underbrace{\sum_{j=1}^n \mathbb{E}[f(x^*) - f([x^*]_j)]}_{A_1} + \underbrace{\sum_{j=1}^n \mathbb{E}[f([x^*]_j) - U_j([x^*]_j)]}_{A_2} + \underbrace{\sum_{j=1}^n \mathbb{E}[U_j([x^*]_j) - f(x_j)]}_{A_3} && (41)
\end{aligned}$$

In order to make use of relevant results in Kandasamy et al. (2018), we adopt the construction of a finite data-independent grid, $\nu_j \subset \mathcal{X}$, at each step j. Using this, we define $[\cdot]_j : \mathcal{X} \mapsto \nu_j$, such that $[x]_j$ is the closest point to x in ν_j. Next, A_1, A_2 and A_3 are bounded separately, where the bounds on A_1 and A_2 are directly from Kandasamy et al. (2018). For A_1 we have

$$A_1 \leq \sum_{j=1}^n \mathbb{E}[|f(x^*) - f([x^*]_j)|] \leq \sum_{j=1}^n \frac{1}{2j^2} \leq \frac{\pi^2}{12}, \tag{42}$$

where the second inequality follows from Lemma 12 of Kandasamy et al. (2018). For A_2 we have

$$\begin{aligned}
A_2 \leq\ & \mathbb{E}\Big[\sum_{j=1}^n \mathbf{1}\{f([x^*]_j) > U_j([x^*]_j)\} \cdot (f([x^*]_j) - U_j([x^*]_j))\Big] && (43)\\
\leq & \sum_{j=1}^n \sum_{x \in \nu_j} \mathbb{E}\big[\mathbf{1}\{f(x) > U_j(x)\} \cdot (f(x) - U_j(x))\big] \leq \sum_{j=1}^n \sum_{x \in \nu_j} \frac{1}{j^2\sqrt{2\pi}|\nu_j|} = \frac{\sqrt{2\pi}}{12}, && (44)
\end{aligned}$$

where we sum only the positive terms in the first step, bound each term j by the sum of corresponding terms on the grid ν_j and then apply Lemma 13 from Kandasamy et al. (2018). The final equality follows from the cardinality of ν_j, as per its definition in Kandasamy et al. (2018).

Our key contribution is the decomposition in Equation (40) and the following bound on the resulting A_3. We make this decomposition to invoke Equation (32), which defines our proposed asynchronous BO policy. We first bound

$$A_3 \leq \sum_{j=1}^{n} \mathbb{E}[U_j(x_j) - f(x_j)] \tag{45}$$

$$=: A_3', \tag{46}$$

since by the definition of the acquisition rule $U_j([x^*]_j) \leq U_j(x_j)$, regardless of the value x^* (see Equation (32)). We proceed by first simplifying and then bounding A_3'.

$$A_3' = \sum_{j=1}^{n} \mathbb{E}[\mathbb{E}[U_j(x_j) - f(x_j)|\mathcal{D}_j]] = \sum_{j=1}^{n} \mathbb{E}[U_j(x_j) - \mathbb{E}[f(x_j)|\mathcal{D}_j]] \tag{47}$$

$$= \sum_{j=1}^{n} \mathbb{E}[\mu_{\mathcal{D}_j}(x_j) + \sqrt{\beta_j}\,\sigma_{\mathcal{D}_j}(x_j) - \mu_{\mathcal{D}_j}(x_j)] = \sum_{j=1}^{n} \mathbb{E}[\sqrt{\beta_j}\,\sigma_{\mathcal{D}_j}(x_j)] \tag{48}$$

$$\leq \sqrt{\beta_n} \sum_{j=1}^{n} \mathbb{E}[\sigma_{\mathcal{D}_j}(x_j)] \tag{49}$$

$$\leq \sqrt{\beta_n}\ \sqrt{\xi_q} \sum_{j=1}^{n} \mathbb{E}[\sigma_{j-1}(x_j)] \tag{50}$$

$$\leq \sqrt{\beta_n}\ \sqrt{\xi_q}\mathbb{E}\Big[\Big(n \sum_{j=1}^{n} \sigma_{j-1}^2(x_j)\Big)^{1/2}\Big] \tag{51}$$

$$\leq \sqrt{\frac{2\xi_q n \beta_n \Psi_n}{\log(1+\eta^{-2})}} \tag{52}$$

where we (in the following order) use that U_j and thus x_j are fixed given $\mathcal{D}_j$, β_j is non-decreasing, Lemma 10 from Kandasamy et al. (2018), Cauchy-Schwarz and finally Lemma 7 from Kandasamy et al. (2018).

Using the bounds on A_1, A_2 and A_3 we get

$$\mathrm{BCR}(n) \leq \frac{\pi^2 + \sqrt{2\pi}}{12} + \sqrt{\frac{2\xi_q n \beta_n \Psi_n}{\log(1+\eta^{-2})}} \tag{53}$$

from which we can then construct the bound

$$\mathrm{BSR}(n) \leq \frac{\pi^2 + \sqrt{2\pi}}{12n} + \sqrt{\frac{2\xi_q \beta_n \Psi_n}{n \log(1+\eta^{-2})}} = \frac{C_1}{n} + \sqrt{\frac{C_2 \xi_q \beta_n \Psi_n}{n}} \tag{54}$$

using $\mathrm{BSR}(n) \leq \frac{1}{n}\mathrm{BCR}(n)$ from, e.g., (Kandasamy et al., 2018). We define $C_1 \equiv \frac{\pi^2+\sqrt{2\pi}}{12}$ and $C_2 \equiv \frac{2}{\log(1+\eta^{-2})}$, which are independent of both q and n. □

The informal statement in Theorem 1 follows by realizing that the sequence β_j in Equation (36) satisfies $\beta_n \in \mathcal{O}(\log(n))$ and $\frac{1}{n} \in o\Big(\sqrt{\frac{\log(n)}{n}}\Big)$.

For completeness, we state the corresponding bounds on the $\mathrm{BSR}(n)$ under the sequential and asynchronous TS policies, as shown in Kandasamy et al. (2018).

Theorem 3 (Bound on $\mathrm{BSR}(n)$ for sequential TS (Corollary 15, Kandasamy et al. (2018))). *Let $f \sim \mathcal{GP}(0, k_\phi)$, where $k_\phi : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ satisfies Assumption 8 in (Kandasamy et al., 2018) and, w.l.o.g., $k_\phi(x,x) \leq 1$. Then, for*

sequential TS the Bayes simple regret after n queries can be bounded as

$$\mathrm{BSR}(n) \leq \frac{2\pi^2 + \sqrt{2}\pi^{5/2}}{12n} + \sqrt{\frac{2\beta_n \Psi_n}{n \log(1+\eta^{-2})}}. \tag{55}$$

Theorem 4 (Bound on BSR(n) for asynchronous TS (Theorem 14, Kandasamy et al. (2018))). *Let $f \sim \mathcal{GP}(0, k_\phi)$, where $k_\phi : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ satisfies Assumption 8 in (Kandasamy et al., 2018) and, w.l.o.g., $k_\phi(x,x) \leq 1$. Then, for the TS acquisition method used in Algorithm 1, the Bayes simple regret after n queries can be bounded as*

$$\mathrm{BSR}(n) \leq \frac{2\pi^2 + \sqrt{2}\pi^{5/2}}{12n} + \sqrt{\frac{2\xi_q \beta_n \Psi_n}{n \log(1+\eta^{-2})}}. \tag{56}$$

Noting that the bound in Theorem 4 is equal in all relevant aspects to our bound in Theorem 2, all downstream conclusions apply equally to both bounds. In particular, we can conclude that under an uncertainty sampling initialization (Desautels et al., 2014; Krause et al., 2008; Kandasamy et al., 2018), the quantity ξ_q can be bounded by a small kernel-dependent constant, independent of n. We refer to Kandasamy et al. (2018) for further details on this. Finally, we can thus conclude that, under the uncertainty sampling initialization, standard asynchronous UCB is almost as good as sequential Thompson sampling. This immediately follows from Corollary 4 of Kandasamy et al. (2018) and the equivalence of the bounds in Theorems 2 and 4.

C Proof Proposition 1

We begin by restating Proposition 1.

Proposition (Marginalized UCB is the Kriging Believer). *Consider the random Upper Confidence Bound, $\alpha_{UCB}(x|D_n, (\mathcal{B}, \mathbf{y}_b))$, of the GP surrogate posterior, with unobserved function values, $\mathbf{y}_b$, at known busy locations, $\mathcal{B}$. Then it holds that*

$$\mathbb{E}[\alpha_{UCB}(x \mid D_n, (\mathcal{B}, \mathbf{y}_b)) \mid D_n, \mathcal{B}] = \alpha_{UCB}(x \mid D_n, (\mathcal{B}, \mu_b)).$$

Proof.

$$\begin{aligned}\mathbb{E}[\alpha_{UCB}(x \mid D_n, D_b) \mid D_n, \mathcal{B}] &= \int \left[\mu(x \mid D_n, D_b) + \sqrt{\beta}\sigma(x \mid D_n, D_b)\right] \, p(\mathbf{y}_b \mid D_n) \mathrm{d}\mathbf{y}_b \\ &= \int \mu(x \mid D_n, D_b) \, p(\mathbf{y}_b \mid D_n) d\mathbf{y}_b \; + \; \int \sqrt{\beta}\sigma(x \mid D_n, D_b) \, p(\mathbf{y}_b \mid D_n) \mathrm{d}\mathbf{y}_b\end{aligned}$$

We note that the value for $\sigma(x|D_n, D_b)$ (as given in Equation (3)) does not depend on the as-of-yet unknown outputs. This makes the second expectation trivial, allowing the objective to be written as

$$\begin{aligned}\mathbb{E}[\alpha_{UCB}(x \mid D_n, D_b) \mid D_n, \mathcal{B}] = \int k(x, X \cup \mathcal{B}) \left[\begin{bmatrix} K_{XX} & K_{X\mathcal{B}} \\ K_{\mathcal{B}X} & K_{\mathcal{B}\mathcal{B}} \end{bmatrix} + \eta^2 \mathbf{I}_{n+q-1}) \right]^{-1} \begin{bmatrix} \mathbf{y} \\ \mathbf{y}_b \end{bmatrix} \, p(\mathbf{y}_b \mid D_n) \mathrm{d}\mathbf{y}_b \\ + \sqrt{\beta}\sigma(x \mid X, \mathcal{B})\end{aligned}$$

The only random variable in the above integral is $\mathbf{y}_b$, which has known expectation μ_b. This integral therefore exactly recovers the prediction if we assume that we will observe the mean value of the GP for the currently busy locations

$$\begin{aligned}\mathbb{E}[\alpha_{UCB}(x \mid D_n, D_b) \mid D_n, \mathcal{B}] &= \mu(x \mid D_n, (\mathcal{B}, \mu_b)) + \sqrt{\beta}\sigma(x \mid X, \mathcal{B}) \\ &= \alpha_{UCB}(x \mid D_n, (\mathcal{B}, \mu_b)).\end{aligned}$$

□

We note that this is equivalent to the KB heuristic (Ginsbourger et al., 2010).

D Implementation details

We implement our optimization pipeline in BoTorch (Balandat et al., 2020) and GPyTorch (Gardner et al., 2018), and make the code available at https://github.com/fortuinlab/SAIS-BO. A zero-mean Gaussian Process prior with ARD-RBF kernel is used to form the surrogate in all experiments. The kernel hyperparameters, ϕ, and the observation noise, η^2, were fit to optimize the posterior $p(\phi|\mathbf{y})$ (Williams and Rasmussen, 2006). We use the dimensionality-scaled hyper-prior on lengthscales proposed by Hvarfner et al. (2024), which is the default in BoTorch. All inputs were normalized to the unit hypercube $[0,1]^d$, and function values were standardized to have zero mean and unit variance.

The optimization of the acquisition function, as well as that of marginal likelihood, was carried out using a multi-restart strategy with the L-BFGS-B algorithm (Byrd et al., 1995). Following standard practice, we optimize the best 10 from an initial set of $1000d$ candidates. We deem this to be sufficient to exclude the optimization of the acquisition function as a source of the observed effects.

We kickstart the optimization with $3 * d$ initial data points and then initialize the q workers. The input locations of initial data, as well as the first batch of q workers, are drawn from a randomly perturbed Halton sequence in the appropriate dimension (Halton, 1964).

The Lipschitz penalizers of LP-UCB and LLP-UCB are approximated in a differentiable manner using $p = -5$, as suggested by Alvi et al. (2019), and use $\gamma = 1$ (Appendix A.2).

The implementation of AEGIS is adopted from the authors De Ath et al. (2021a), where the approximate Pareto front is found with code from https://github.com/georgedeath/aegis/blob/main/aegis/batch/nsga2_pygo.py.

We use $N = 500$ *Monte Carlo* samples to estimate the expected EI, since this is the BoTorch default (https://botorch.org/docs/acquisition).

E Toy experiments

A number of additional toy experiments are presented.

1−d task Figure E.1 shows the continuation of Figure 3. It can be seen that the trend of the standard UCB and its purpose-built variants querying essentially the same locations, even though the UCB does not take into account the busy locations, persists.

2−d task Figure E.2 shows additional query trajectories for standard acquisition, as well as *penalization*- and *randomness*-based methods. Rows 1 and 2, as well as 3 and 4, respectively, share initial data and initial worker locations. It can be seen that the standard methods (UCB, LogEI) gradually transition from an exploration to an exploitation phase and get close to the optimum with the iteration budget. Purpose-built methods do not display this transition from exploration to exploitation and do not generally find the optimum.

F Optimization tasks

F.1 Synthetic test functions

The noiseless synthetic test functions we use are available as part of the BoTorch package (Balandat et al., 2020). We present experiments in varying dimensions for the following functions: Ackley, Hartmann, Egg Holder, Michalewicz, and Rosenbrock. Please refer to https://botorch.readthedocs.io/en/latest/test_functions.html#module-botorch.test_functions.synthetic or https://www.sfu.ca/~ssurjano/optimization.html for more details.

F.2 Real-world tasks

PROTEUS PROTEUS is a coupled atmosphere-interior framework to simulate the temporal evolution of rocky planets (Lichtenberg et al., 2021; Nicholls et al., 2024). This deterministic forward simulator takes in an

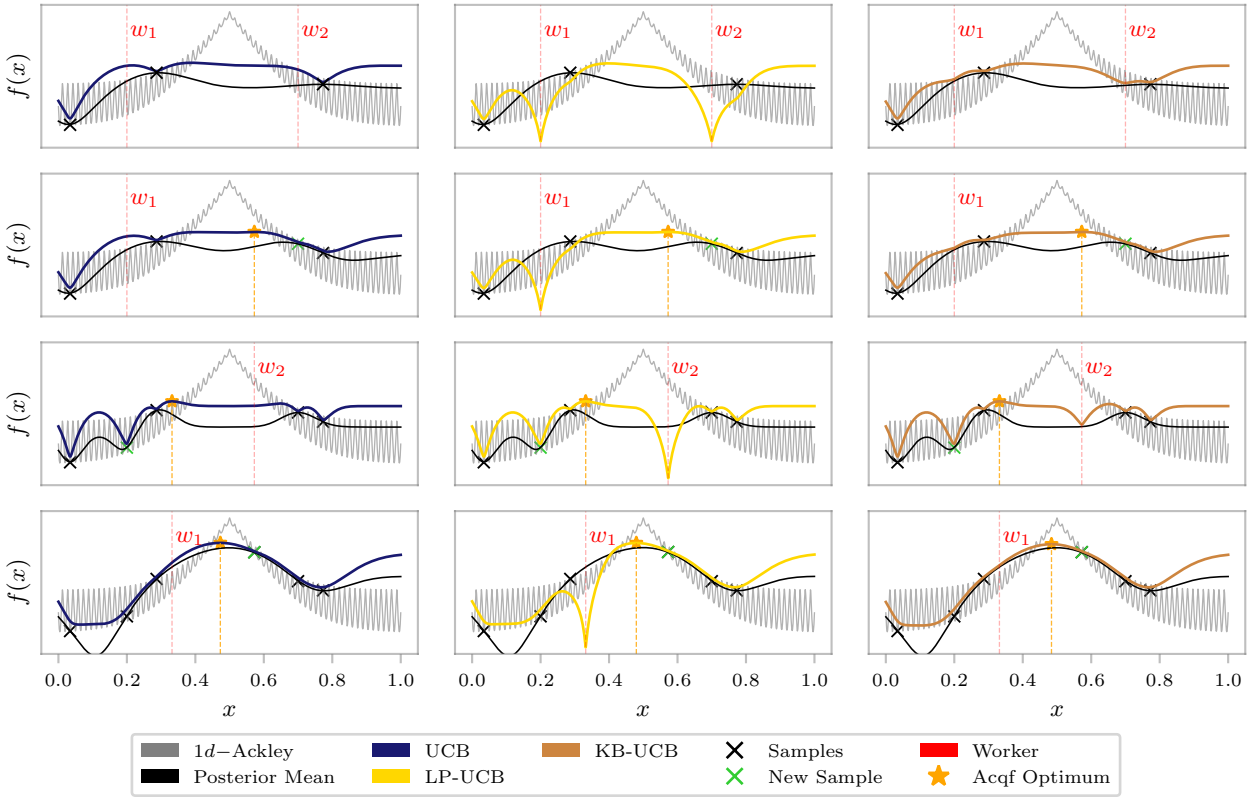

Figure E.1: Continuation of Figure 3. It can be seen that the standard UCB queries essentially the same points as purpose-built methods, without taking into account the busy location. The update via the new sample provides sufficient information to prevent redundant or even repeated queries.

initial condition, x, returning observables $\gamma = PROTEUS(x)$. Given observables γ_0, one may want to infer the associated initial condition, i.e., perform a point-wise inversion of $PROTEUS(\cdot)$ at γ_0. We frame this inference as an optimization problem, with

$$\min_{x \in \mathcal{X}} \|PROTEUS(x) - \gamma_0\|^2, \tag{57}$$

with known optimum $(x^*, 0)$ for x^* such that $PROTEUS(x^*) = \gamma_0$. In this work, nine input variables are considered. Due to its adaptive time stepping and resulting varying evaluation times, PROTEUS naturally lends itself to asynchronous BO. PROTEUS is freely available, and we refer to https://fwl-proteus.readthedocs.io/en/latest/ for detailed instructions on the installation.

Pest control In this benchmark, the aim is to minimize the spread of pests and the (monetary) cost expended to this end (Oh et al., 2019). The design space consists of 25 categorical variables with five levels each ($\approx 2.98 \times 10^{17}$ combinatorial choices). This space represents 25 stations and the amount of pesticide used at each. We treat this combinatorial problem as an ordinal one, by discretizing the input space $[0, 1]^{25}$ to $\{1, 2, 3, 4, 5\}^{25}$, thus creating a step function. This allows the categorical problem to be solved with standard continuous input methodology. The asynchronicity is provided by the varying cost throughout the input space, which we take to be the evaluation time. In order to compute the $\log(R)$, we take -12 to be the optimal value. The original code we use for this is from https://github.com/yucenli/bnn-bo and https://github.com/QUVA-Lab/COMBO.

XG-Boost Hyperparameter optimization is a task every ML practitioner faces. The choice of hyperparameters often makes or breaks a model. For XG-Boost, we consider nine hyperparameters, which we tune to optimize the 5-fold cross-validation accuracy. This score is computed from the classification performance on the UCI Breast Cancer data (Wolberg et al., 1993). The hyperparameters optimized are learning rate, number of boosting rounds (trees), maximum tree depth, minimum loss reduction to make a split, fraction of training examples to grow

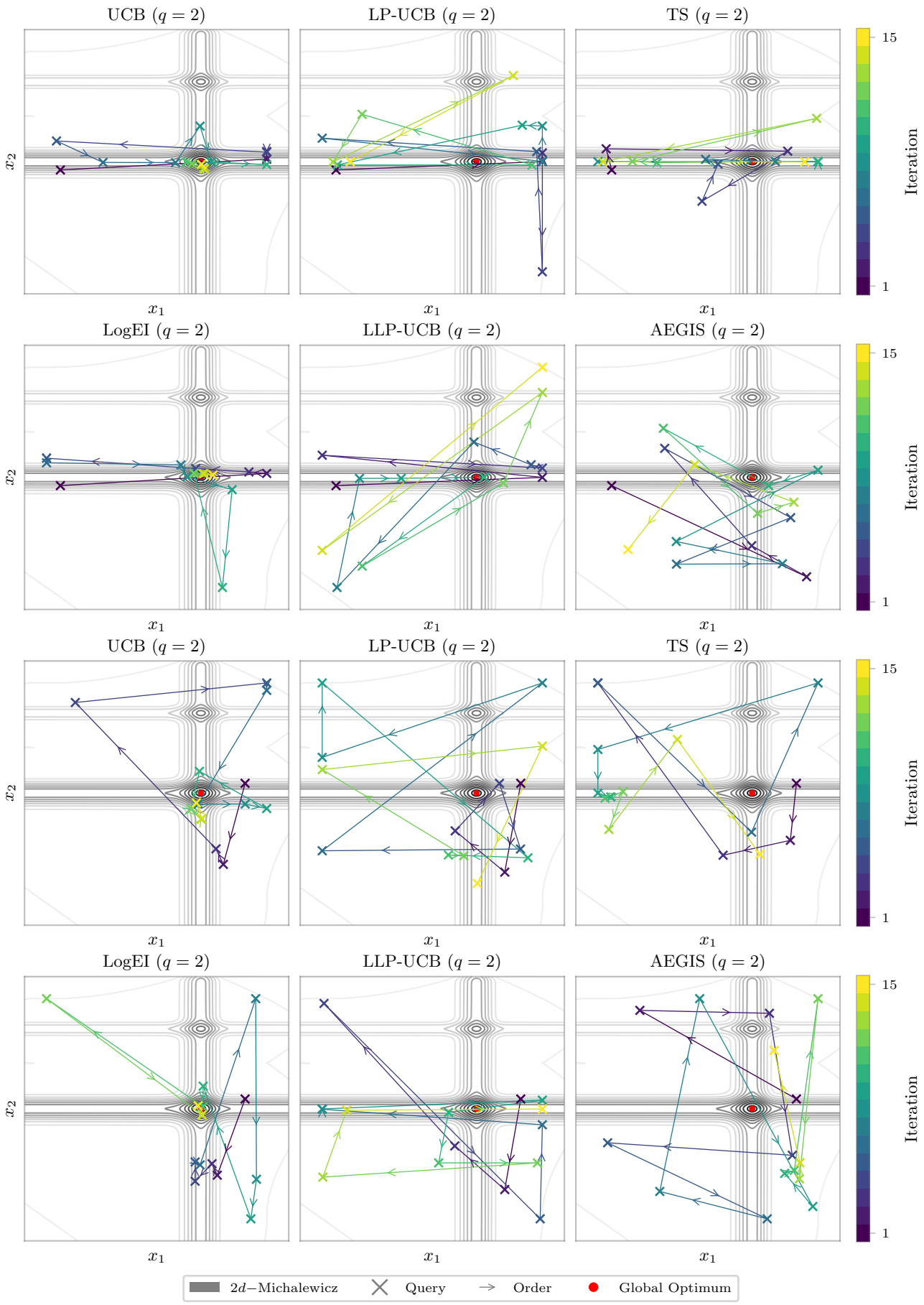

Figure E.2: Query trajectories, see Appendix E for a detailed description.

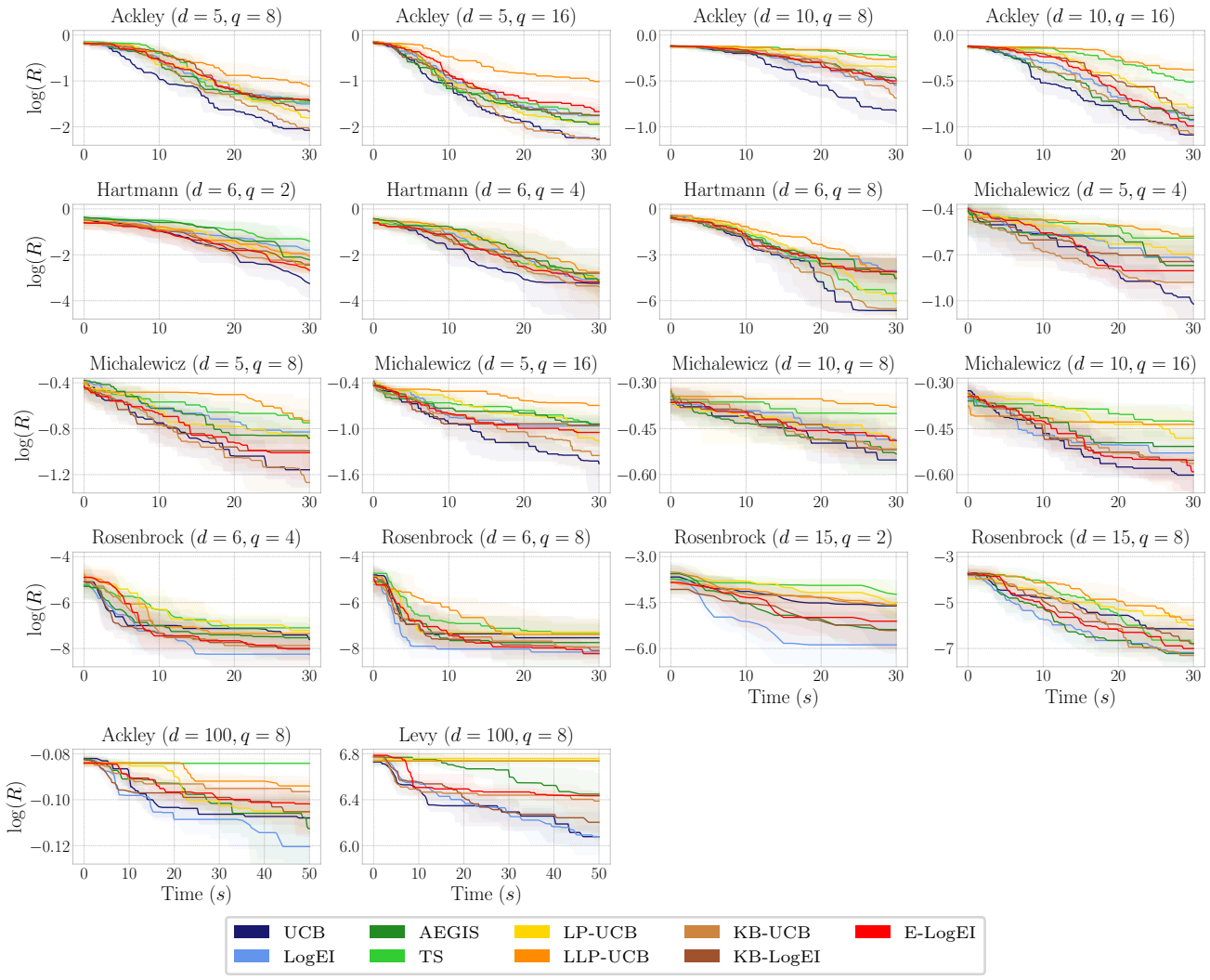

Figure G.1: Standard acquisition functions are not significantly outperformed on any of our synthetic test functions. In fact, for our experiments, UCB often outperforms alternatives designed for asynchronous BO.

each tree on, fraction of features to use per tree, fraction of features to use per node (split), as well as the L1 and L2 regularization parameters on the leaf weights. The training time depends, e.g., on the maximum tree depth, resulting in the desired heterogeneous function evaluation times. The optimal value of the cross-validation accuracy is known to be 1, allowing for computation of the $\log(R)$.

CNN Arguably the most widely used data set in image classification, CIFAR10 (Krizhevsky and Hinton, 2009) contains $60,000$ 32x32 pixel images in ten classes. We randomly select a training and a validation set of $25,000$ examples each. In our experiment, we learn the hyperparameters of a 6-layer CNN pipeline to optimize validation accuracy after 20 epochs of training. This is framed as a 9-dimensional optimization problem of batch size, learning rate, momentum, and filter sizes for the six filters. At a fixed number of epochs, filter and batch size directly affect the train time of the CNN, creating varying evaluation times throughout the input space. The optimal value of the validation accuracy is known to be 1, allowing for computation of the $\log(R)$.

G Additional experimental results

In Figures G.1 to G.5, we give additional experimental results on variants of the problems presented in Section 5. Additionally, we present a quantitative assessment in the form of a win-rate measure and a Mann-Whitney

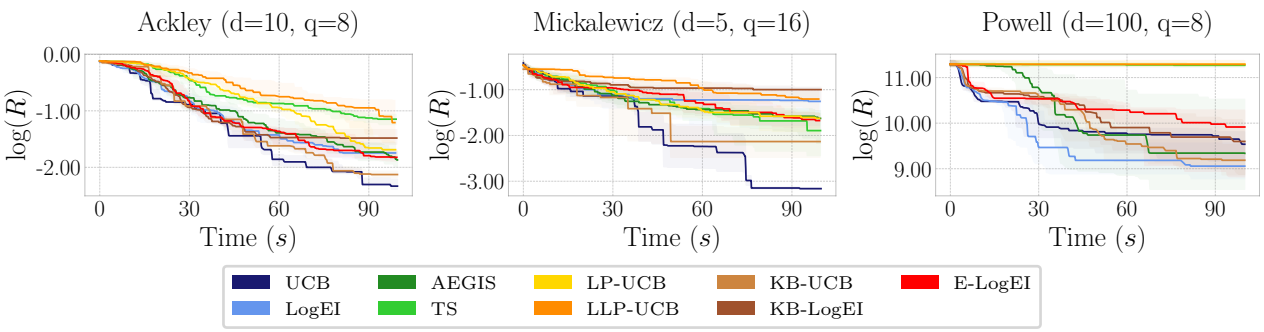

Figure G.2: Extended run times show qualitatively the same results.

U-test. Tables G.1 and G.2 show the win-rate results and should be read as "row-method" was better than "column-method" a fraction "entry" of the 20 trials. It can be seen that this is always larger than 0.5 for the standard UCB, and often even close to 1.0. Tables G.3 and G.4 show the p-values of a Mann-Whitney U-test, with the null hypothesis: The regret values after $t = 30$s of "row-method" and "column-method" follow the same distribution. As can be seen from the tiny p-values, the null can almost always be rejected at very small significance levels.

Table G.1: Ackley $(d = 10, q = 8)$, win-rate @ $t = 30$s

	LogEI	TS	LP-UCB	LLP-UCB	AEGIS	KB-LogEI	KB-UCB	E-LogEI
UCB	0.85	1.0	0.95	1.0	1.0	0.95	0.75	0.9
LogEI	–	0.9	0.7	0.85	0.6	0.55	0.4	0.4

Table G.2: Michalewicz $(d = 5, q = 16)$, win-rate @ $t = 30$s

	LogEI	TS	LP-UCB	LLP-UCB	AEGIS	KB-LogEI	KB-UCB	E-LogEI
UCB	0.75	0.85	0.75	1.0	1.0	0.85	0.55	0.8
LogEI	–	0.6	0.45	0.8	0.6	0.5	0.2	0.4

Table G.3: Ackley $(d = 10, q = 8)$, MWU @ $t = 30$s

	LogEI	TS	LLP-UCB	AEGIS	KB-LogEI	KB-UCB	E-LogEI
UCB	1.79×10^{-4}	1.06×10^{-7}	9.17×10^{-8}	9.13×10^{-7}	9.75×10^{-6}	2.07×10^{-2}	6.67×10^{-6}
LogEI	–	1.44×10^{-4}	5.09×10^{-4}	5.98×10^{-1}	9.68×10^{-1}	3.15×10^{-2}	9.03×10^{-1}

Table G.4: Michalewicz $(d = 5, q = 16)$, MWU @ $t = 30$s

	LogEI	TS	LLP-UCB	AEGIS	KB-LogEI	KB-UCB	E-LogEI
UCB	1.12×10^{-3}	8.29×10^{-5}	4.54×10^{-7}	5.87×10^{-6}	2.60×10^{-5}	3.65×10^{-1}	2.75×10^{-4}
LogEI	–	5.61×10^{-1}	4.60×10^{-4}	3.37×10^{-1}	7.15×10^{-1}	1.14×10^{-2}	5.43×10^{-1}

H Compute

All experiments were conducted on a single NVIDIA V100 GPU with 32 GB of memory. Each synthetic experiment required approximately 8 hours, while the real-world tasks ranged from about 8 hours (Pest Control) to 48 hours (CNN). Naturally, this was preceded by many hours of testing and prototyping.

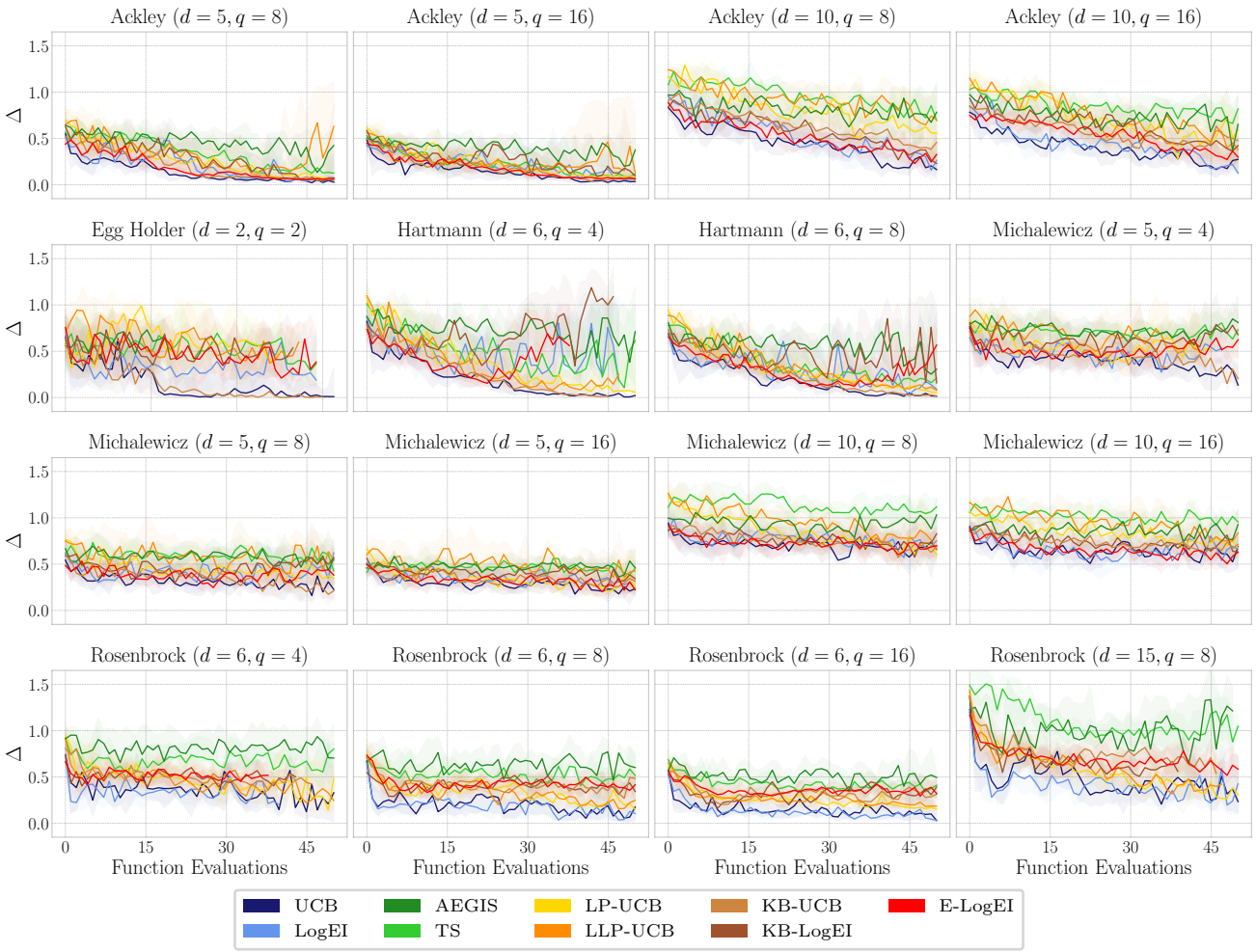

Figure G.3: Standard acquisition functions query closer to busy locations than alternatives designed for asynchronous BO, but do not systemically repeat queries.

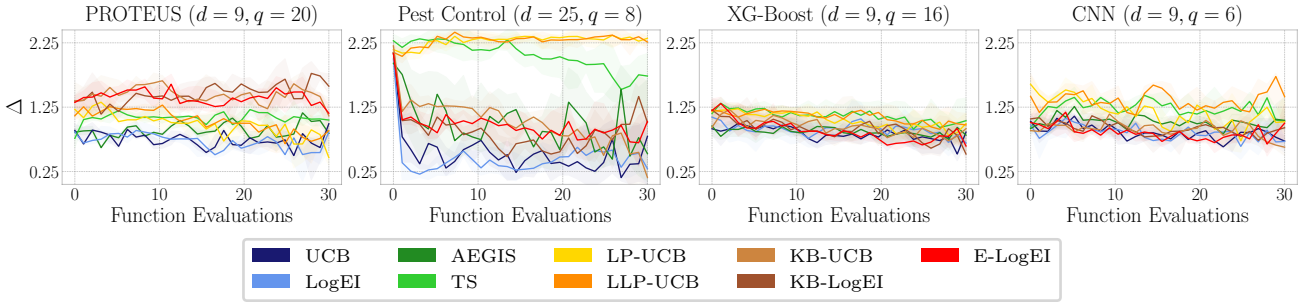

Figure G.4: While standard acquisition queries the closest to busy locations, distances on real-world tasks are significantly larger than zero for all methods.

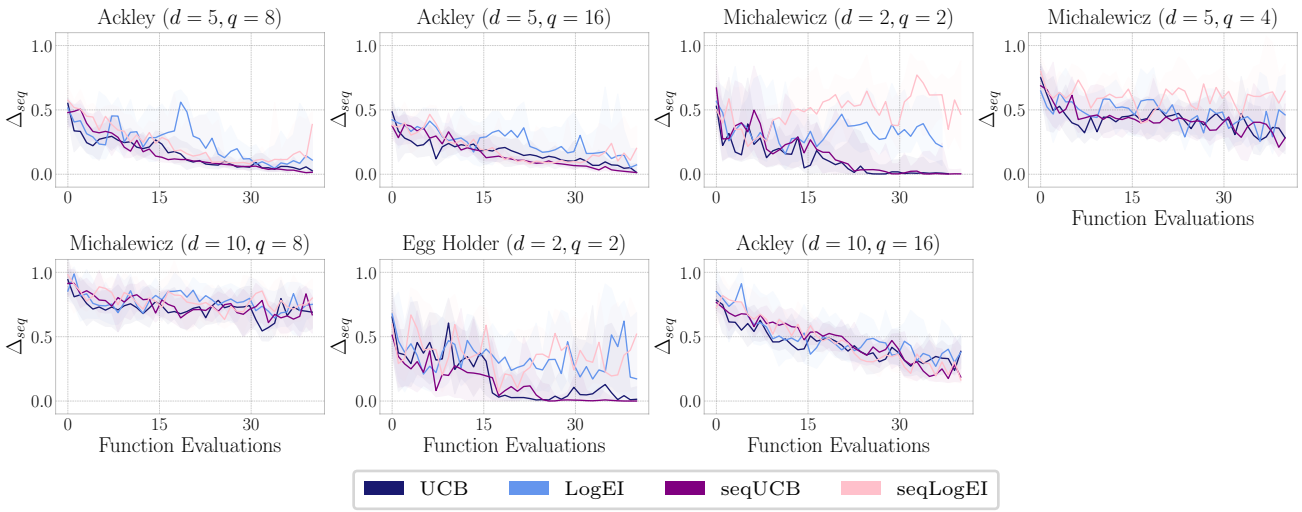

Figure G.5: Standard acquisition functions mimic the distance profile of their respective sequential counterparts.