Rethinking Exploration In Asynchronous Bayesian Optimization: Standard Acquisition Is All You Need

Ben Riegler
TU Munich, Helmholtz AI
ben.riegler@tum.de

James Odgers
Helmholtz AI
james.odgers@helmholtz-munich.de

Vincent Fortuin
TU Munich, Helmholtz AI
vincent.fortuin@tum.de

Abstract

Asynchronous Bayesian optimization is widely used for gradient-free optimization in domains with independent parallel experiments and varying evaluation times. Previous works posit that standard acquisitions lead to under exploration of the space via redundant queries. We show that this is not the case: standard acquisition functions avoid redundant queries thanks to the intermediate posterior updates. We show theoretically that *penalization*-based methods are approximations to the Kriging Believer, a method with known shortcomings. By analysing distance to busy locations, we also show that by enforcing diversity incumbent methods over-explore and under-exploit in asynchronous settings, reducing their performance. In contrast, our extensive experiments demonstrate that simple standard acquisition functions, like the Upper Confidence Bound, match or outperform purpose-built asynchronous methods across synthetic and real-world tasks.

1 Introduction

The exploration-exploitation trade-off is fundamental to many learning systems that must efficiently allocate limited resources between gathering new information and leveraging existing knowledge [5, 36]. One area this challenge arises in is black-box optimization (BBO), where the goal is to optimize functions with no known analytical form or gradient information via expensive to run experiments. Bayesian optimization (BO) provides a principled framework for this exploration-exploitation balance, using probabilistic surrogate models to guide the search toward promising regions. Applications span materials discovery [8, 32], chemical design [19, 39], nuclear and accelerator physics [11, 33], and hyperparameter tuning in machine learning [7, 24, 34, 43].

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 [10, 13, 25, 38, 44]. 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 fails at exploration 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.*

First Exploration in AI Today Workshop at ICML (EXAIT at ICML 2025).

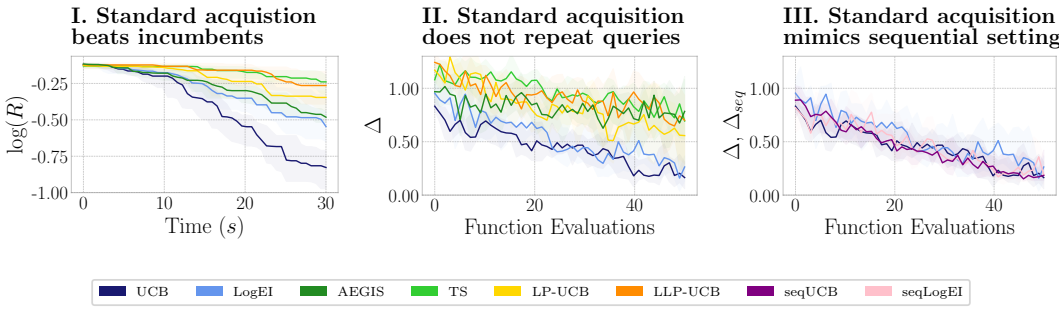

Figure 1: We find that (I.) standard acquisition outperforms or matches methods purpose-built for asynchronous Bayesian optimization. (II.) Comparing the distances of queries to the closest busy location, we see that standard acquisition exhibits the desirable transition from exploration to exploitation, but does not repeat queries. (III.) Standard acquisition functions query at similar distances as their sequential counterparts, suggesting that nearby sampling is part of a desirable exploration-exploitation trade-off. Results shown on Ackley $(d = 10, q = 8)$.

All existing works on asynchronous BO—explicitly or implicitly—build on Hypothesis 1, by proposing acquisition rules enforcing diversity in the asynchronous queries. This is generally achieved through either *randomness* [9, 23], or through *penalization* of the acquisition function at and around busy locations [1]. Earlier work approximately marginalizes out the unknown function values via Monte Carlo sampling [17, 21, 34], or replaces them with a hallucinated proxy [16].

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 are unnecessary. We observe that Hypothesis 1 fails to account for the update to the surrogate model with a new datum, before the next input is asynchronously selected. Additionally, we show a novel connection between *penalization*-based methods and the well-known Kriging Believer (KB) [16] heuristic. Given the known shortcomings of the KB [1, 9], we argue that the poor empirical results of *penalization*-based methods are unsurprising [9].

Instead, we propose to use existing standard and seemingly naive acquisition rules, such as the Upper Confidence Bound (UCB) [3, 35] or Expected Improvement (EI) [29]. Excellent performance on an extensive suite of experiments supports our conceptual and theoretical 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.

- We identify a conceptual flaw in reasoning about exploration needs in asynchronous BO, showing that Hypothesis 1 fails to account for information gained from completed evaluations.
- We demonstrate theoretically that diversity-enforcing methods approximate the Kriging Believer heuristic, explaining their poor exploration-exploitation balance.
- We show empirically that standard acquisition functions achieve effective exploration in asynchronous BO, matching or outperforming purpose-built methods across synthetic and real-world tasks.
- We provide evidence that standard approaches naturally balance exploration and exploitation similarly to optimally informed sequential BO, suggesting explicit diversity enforcement may actually harm effective exploration.

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 $x_i \in \mathcal{X}$, resulting in a time-delayed observation of the corresponding

output $y_i \in \mathbb{R}$. 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 [22]. 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, proposing the next query location $x' \in \mathcal{X}$. See Appendix A for an illustration of synchronous vs. asynchronous BO and pseudo code of an asynchronous BO routine.

Gaussian Process surrogate A zero-mean Gaussian Process (GP) prior, $\mathcal{G} = GP(0, k_\phi(\cdot,\cdot))$, is defined through a positive definite covariance function $k_\phi : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$, with hyperparameters ϕ. 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 [40].

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

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

The posterior mean $\mu : \mathcal{X} \mapsto \mathbb{R}$ and variance $\sigma^2 : \mathcal{X} \mapsto \mathbb{R}_{\geq 0}$ are analytically tractable as,

$$\mu(x \mid D_n) = k_\phi(x, X)(K_{XX} + \sigma_y^2 \mathbf{I}_n)^{-1}\mathbf{y} \tag{2}$$

$$\sigma^2(x \mid D_n) = k_\phi(x, x) - k_\phi(x, X)(K_{XX} + \sigma_y^2 \mathbf{I}_n)^{-1} k_\phi(X, x) \tag{3}$$

with K_{XX} being the kernel matrix of input locations X, i.e., $[K_{XX}]_{ij} = k_\phi(x_i, x_j)\ \forall\ i, j \in [n]$.

Acquisition function 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 via a heuristic termed the acquisition function. A number of acquisition functions proposed in the literature are considered in this work and outlined in the following Section 3.

3 Acquisition rules

The probabilistic nature of the surrogate model allows the acquisition function to reason about uncertainty in $f(\cdot)$. This, in turn, guides the trade-off of exploration and exploitation of 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

$$x' = \arg\max_{x \in \mathcal{X}} \alpha(x \mid D_n). \tag{4}$$

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 more detailed discussion can be found in Appendix B.

3.1 Standard acquisition functions

Upper Confidence Bound Encoding the exploration-exploitation trade-off via the parameter β, the GP Upper Confidence Bound (UCB) [3, 35] is defined as

$$\alpha_{UCB}(x) = \mu(x \mid D_n) + \sqrt{\beta}\sigma(x \mid D_n). \tag{5}$$

Expected Improvement The Expected Improvement (EI) [22, 29] 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,

$$\alpha_{EI}(x) = \mathbb{E}_{f|D_n}[\max(f(x) - y_n^*, 0)]. \tag{6}$$

For numerical stability in the optimization, this work uses the natural logarithm of EI [2].

3.2 Asynchronous acquisition rules

Building on Hypothesis 1, numerous methods for asynchronous batch Bayesian optimization have been proposed in previous works. These approaches generally fall into several categories: Monte Carlo sampling methods that approximately marginalize over unknown function values at busy locations [17, 21, 34], hallucination-based methods like the Kriging Believer which replace unknown values with assumed values [15, 16], *randomness*-based approaches including Thompson sampling [23, 37] and AEGIS [9], and *penalization*-based methods that down-weight acquisition functions near busy locations using (local) Lipschitz estimates [1]. We note that these existing works all either did not compare their method to standard acquisition or did not outperform it. Detailed descriptions of these methods are provided in Appendix B.2.

4 Conceptual and theoretical analysis

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., Halton sequence [20]) in the search space, $\mathcal{X}$, guaranteeing initial diversity. Once a worker finishes its function evaluation, y_{n+1}, at x_{n+1}, it immediately makes a new acquisition. 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, as outlined in Section 3, by

$$x' = \arg\max_{x \in \mathcal{X}} \alpha(x \mid D_{n+1}). \tag{7}$$

Observation 1 (Hypothesis 1 neglects GP surrogate update)**.** *Hypothesis 1 states that the new input location, x', will either be redundant or even repeated. However, the new datum, (x_{n+1}, y_{n+1}), updates the GP surrogate, resulting in a very different optimum for the acquisition function.*

While Observation 1 suggests that there should be some exploration from standard acquisition in Asynchronous BO, it does not give guarantees that new queries will not be arbitrarily close to running experiments. However, this does not mean that standard acquisition will result in a bad exploration-exploitation trade-off. For instance, in the sequential BO setting, to the best of our knowledge, there is no theoretical guarantee that sequential queries will not be arbitrarily close to recent queries—yet in that setting it has been empirically verified that standard acquisition will result in good exploration-exploitation trade-offs. Just as in the sequential case, each asynchronous acquisition is preceded by a newly completed evaluation. This should lead to a shift in the acquisition function of comparable magnitude, and thus to similarly diverse queries. We verify this empirically in Section 5.2.

4.2 Marginalizing busy locations in the UCB approximates the Kriging Believer and is suboptimal

The *penalization*-based methods by Alvi et al. [1] (Section 3.2) are derived as an approximation of the acquisition function with the unknown function values, $\mathbf{y}_b = \{y_j\}_{j=1}^{q-1}$, at the busy locations, $\mathcal{B} = \{x_j\}_{j=1}^{q-1}$, marginalized out. Formally, their method is proposed to approximate

$$\alpha_{LP}(x \mid \mathcal{B}) \approx \mathbb{E}[\alpha(x \mid D_n, D_b) \mid D_n, \mathcal{B}] \tag{8}$$

$$= \int \alpha(x \mid D_n, D_b)\, p(\mathbf{y}_b \mid D_n) d\mathbf{y}_b, \tag{9}$$

with $D_b = (\mathcal{B}, \mathbf{y}_b)$ being the unknown (i.e., random) data from busy locations. This expectation is taken under the GP posterior predictive, such that $\mathbf{y}_b$ simply follows a multivariate Gaussian distribution. In particular, following Equations (2) and (3), we have,

$$p(\mathbf{y}_b \mid D_n) = \mathcal{N}(\mathbf{y}_b \mid \mu_b, \Sigma_b), \tag{10}$$

with

$$\mu_b = K_{\mathcal{B}X}(K_{XX} + \sigma_y^2 \mathbf{I}_n)^{-1}\mathbf{y} \in \mathbb{R}^{q-1} \tag{11}$$

$$\Sigma_b = K_{\mathcal{B}\mathcal{B}} - K_{\mathcal{B}X}(K_{XX} + \sigma_y^2 \mathbf{I}_n)^{-1} K_{X\mathcal{B}} \in \mathbb{R}^{(q-1)\times(q-1)}. \tag{12}$$

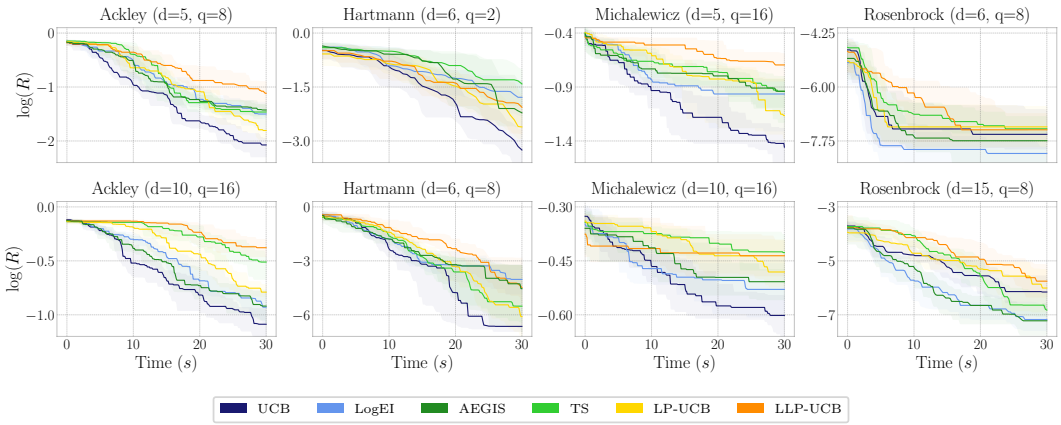

Figure 2: **Standard acquisition dominates across dimensions and number of workers.** The standard UCB outperforms all other methods on all tasks, with the exception of the Rosenbrock function, where standard LogEI performs better. This superior performance is robust to the dimensionality and the number of workers, whereas the performance of the purpose-built asynchronous methods is always worse and more sensitive to task parameters.

The intractability of the expectation in Equation (8) for the EI is what motivated Ginsbourger et al. [17], who approximately marginalize out the unknown function values via a Monte Carlo sampling-based approach (Section 3.2). But, in the case of the UCB, the acquisition function used by *penalization*-based methods [1], this integral is in fact analytically tractable and leads to a well-known heuristic.

Proposition 1 (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*

$$\mathbb{E}[\alpha_{UCB}(x \mid D_n, (\mathcal{B}, \mathbf{y}_b)) \mid D_n, \mathcal{B}] = \alpha_{UCB}(x \mid D_n, (\mathcal{B}, \mu_b)). \tag{13}$$

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 $\sigma(x \mid D_n, D_b)$ 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 heuristic well known as the Kriging Believer [16]. Proposition 1 also gives an additional insight into the Kriging Believer. We have now shown that, in the case of the UCB, the Kriging Believer is maximizing the expected acquisition function. This insight suggests that maximizing the expected value of the acquisition function can ignore the uncertainty of unknown function values, $\mathbf{y}_b$, thereby ignoring promising areas that deserve more exploration. This provides a theoretical underpinning for the poor performance of these methods [1, 9], which we demonstrate in the following section.

5 Experimental evaluation

5.1 Bayesian Optimization Performance

In our results, 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 E.

Synthetic test functions We compare acquisition rules on synthetic test functions across varying dimensions and worker counts, with each experiment repeated 20 times using different initializations

[1]We make our code available at https://github.com/Ben-Riegler/AsyncBO_EXAIT

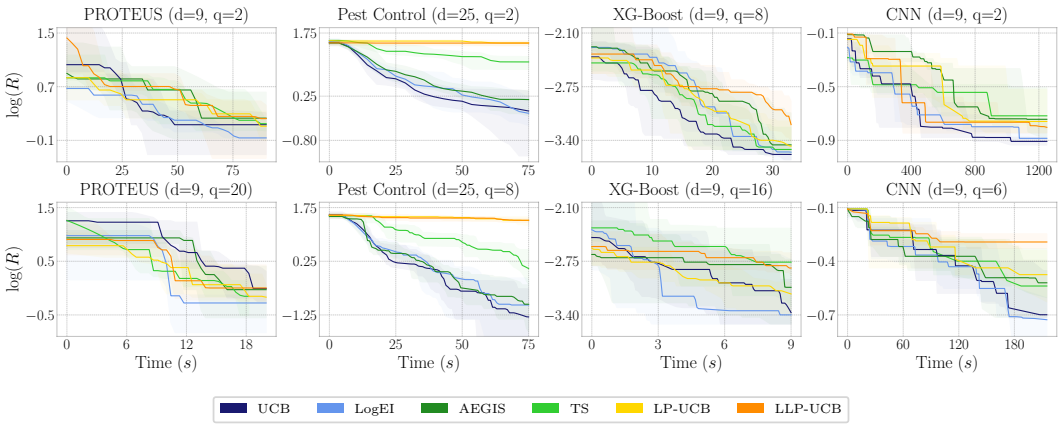

Figure 3: **Standard acquisition performs best on real-world tasks.** Our findings from the synthetic test functions regarding the superior performance of standard acquisition over specialized asynchronous methods transfer to challenging real-world tasks, such as hyperparameter tuning, even in the cases of high dimensionality (e.g., Pest Control) or a large number of workers (e.g., PROTEUS). Moreover, increasing the number of workers appears to disproportionately benefit the standard acquisition methods, particularly on hyperparameter tuning tasks.

and evaluation times sampled from a half-normal distribution with scale parameter $\theta = \sqrt{\pi/2}$ to simulate asynchronous settings [1, 9]. As shown in Figure 2, no *penalization*- or *randomness*-based method outperforms both standard UCB and LogEI, with UCB demonstrating strong performance on nearly all synthetic tasks (except Rosenbrock, where LogEI dominates) despite using a fixed exploration parameter $\beta = 2$ rather than more sophisticated scheduling approaches [35]. While the local Lipschitz *penalization*-based method (LLP-UCB) and Thompson sampling show consistently poor performance, AEGIS occasionally matches standard acquisition performance on specific functions where its predefined exploration-exploitation trade-off (encoded in ϵ_T and ϵ_P) happens to be appropriate—though this fails to generalize across all test functions. Additional results on more test functions and configurations (d, q) can be found in Appendix F.

Real-world tasks To demonstrate the robustness of our findings, we perform experiments on four relevant real-world tasks with 9 different initializations (reduced from 20 due to computational constraints): PROTEUS, an astrophysics simulator requiring optimization of $d = 9$ inputs to match simulator output with observations [28, 30]; Pest Control, a well-known benchmark with 25 categorical variables representing pesticide amounts on crops [31]; and two 9-dimensional hyperparameter tuning tasks using XG-Boost [12] and Convolutional Neural Networks [27]. As shown in Figure 3, standard acquisition functions benefit disproportionately from increased worker counts q compared to existing methods, with this effect particularly pronounced for hyperparameter tuning tasks—contrary to Hypothesis 1, standard methods effectively utilize large numbers of parallel workers without requiring explicit regularization via *penalization* or *randomness*. Consistent with our theoretical predictions in Section 4.2, *penalization*-based approaches show particularly poor performance, especially the local Lipschitz variant on Pest Control, XG-Boost, and CNN tasks.

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 behaviour, 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, i.e., the points which would be busy if asynchronous BO had perfect information. 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, as

$$\Delta \equiv \min_{x_j \in \mathcal{B}} \|x' - x_j\|_2, \tag{14}$$

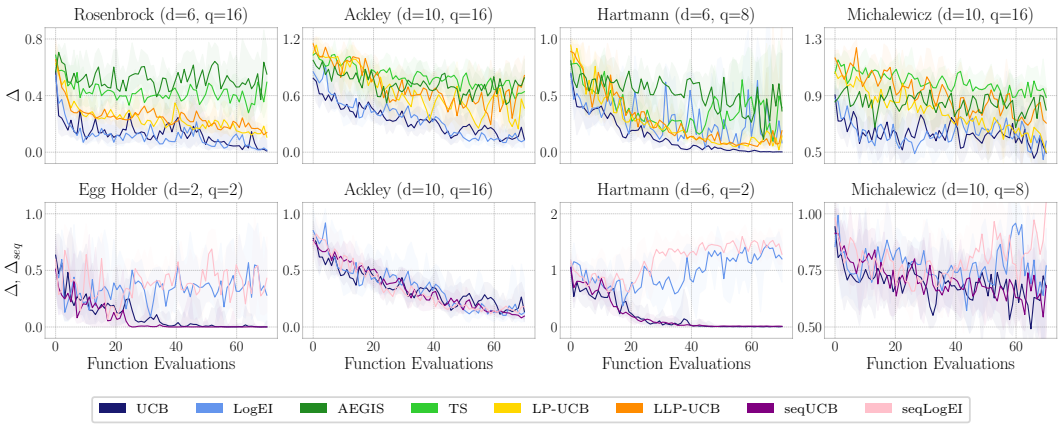

Figure 4: **Standard acquisition functions perform an exploration-exploitation trade-off similar to that of their optimally informed sequential counterparts.** We consider the distance, Δ, of an asynchronous query to the closest of $q-1$ busy locations, and the distance, Δ_{seq}, of a sequential query to the closest out of the $q-1$ previous queries. It can be seen that standard acquisition methods do not repeat queries, and that standard acquisitions query at similar distances in the asynchronous setting as in the sequential setting.

and the distance, Δ_{seq}, of the n^{th} query in sequential BO to the last $q-1$ sequential queries as

$$\Delta_{seq} \equiv \min_{i \in [q-1]} \|x_n - x_{n-i}\|_2. \tag{15}$$

In Figure 4, we show the median and inter-quartile range of these distances for a number of test functions. It can be seen that existing methods query farther from busy locations than standard acquisition, as is intended. But, contrary to Hypothesis 1, standard acquisition does not perform repeated queries at the start of the optimization. In fact, standard acquisition functions often exhibit the desirable transition from an initial exploratory phase (Δ large) to an exploitation stage (Δ small).

Even more interestingly, the comparison of standard acquisition functions to their respective sequential counterparts (Figure 4) shows that standard acquisition in the asynchronous setting closely mimics the query distances of the sequential one. This analysis reveals that the exploration-exploitation behaviour of standard acquisition functions is not dependent on the asynchronous setting, but simply a characteristic of the acquisition functions themselves.

This sheds light on the superior performance of standard acquisition in asynchronous BO shown in Section 5.1. Existing purpose-built methods enforcing diversity through *penalization* or *randomness* seem to suffer from over-regularization, preventing the methods from leaving the exploration phase. On the other hand, standard acquisition performs an exploration-exploitation trade-off closely aligned with that of the optimally informed sequential BO.

6 Discussion and conclusion

Good results in BO require making good exploration-exploitation trade-offs [35]. Hypothesis 1 claims a severe over-exploitation (including repeated queries) from the very start of the optimization for standard acquisition functions in asynchronous BO. However, we point out that the newly obtained evaluation has the potential to result in an acquisition function with a very different optimum (Observation 1). Additionally, we show that standard acquisition methods mimic the exploration-exploitation trade-off of the optimally informed sequential BO. Contrary to Hypothesis 1, it seems that methods explicitly enforcing diversity in queries suffer from over-exploration instead of performing the desired transition from an initial exploration phase to a later exploitation stage.

While we demonstrate in this work that 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. Clearly, knowledge of function values at busy locations should improve the acquisition. In future research, we plan to further explore mechanisms taking into account the busy locations and higher-order moments of the unknown function values to improve performance.

Acknowledgments and Disclosure of Funding

We thank Tommy Rochussen for helpful comments. Vincent Fortuin was supported by a Branco Weiss Fellowship.

References

[1] Ahsan S Alvi, Binxin Ru, Jan Calliess, Stephen J Roberts, and Michael A Osborne. Asynchronous batch Bayesian optimisation with improved local penalisation. *arXiv preprint arXiv:1901.10452*, 2019.

[2] 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.

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

[4] 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.

[5] Donald A Berry and Bert Fristedt. Bandit problems: sequential allocation of experiments (monographs on statistics and applied probability). *London: Chapman and Hall*, 5(71-87):7–7, 1985.

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

[12] Jerome H Friedman. Greedy function approximation: A gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[13] 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.

[14] 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.

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

[16] 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.

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

[18] 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.

[19] 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.

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

[21] Janis Janusevskis, Rodolphe Le Riche, David Ginsbourger, and Ramunas Girdziusas. Expected improvements for the asynchronous parallel global optimization of expensive functions: Potentials and challenges. In *International Conference on Learning and Intelligent Optimization*, pages 413–418. Springer, 2012.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

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

[27] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

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

[33] 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.

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

[35] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

[36] Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[37] 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.

[38] 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.

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

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

[41] 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.

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

[43] 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.

[44] 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.

A Asynchronous BO Algorithm

The formal description of asynchronous BO is given in Algorithm 1. An illustration is also given in Figure 5.

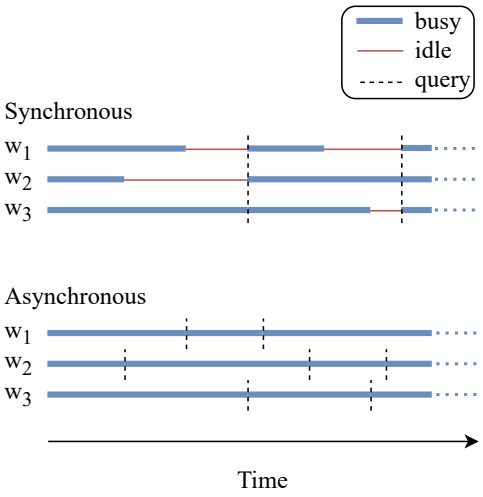

Figure 5: Illustration of Synchronous vs. Asynchronous BO with $q = 3$ workers. The Asynchronous setup allows for more queries in the same wall-clock time by avoiding idle workers.

Algorithm 1 Asynchronous BO with q workers

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

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

B Acquisition function details

B.1 Standard acquisition

Upper Confidence Bound The GP Upper Confidence Bound (UCB) [35] is a simple heuristic based on optimizing a quantile of the credability 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 trade-off is controlled by β, which may be set to a fixed value or according to some schedule [35].

Expected Improvement The Expected Improvement (EI) [22, 29] 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,

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

For numerical stability in the optimization, this work uses the natural logarithm of EI [2].

B.2 Asynchronous acquisition rules

Monte Carlo sampling Ginsbourger et al. [17] 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

$$\alpha_{EEI}(x) = \frac{1}{N}\sum_{i=1}^{N} \alpha_{EI}(x | D_n, (\mathcal{B}, \mathbf{y}_{b,i})), \tag{18}$$

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

$$p(\mathbf{y}_b \mid D_n) = \mathcal{N}(\mathbf{y}_b \mid \mu_b, \Sigma_b), \tag{19}$$

with μ_b and Σ_b as in equations Equations (11) and (12).

While Ginsbourger et al. [17] do this for the EI, this approach may be taken for any analytical acquisition function. But this approach suffers from poor scaling due to the need for repeated posterior sampling. More importantly, this marginalization constitutes a first-moment approximation, rendering it inherently suboptimal, as we argue in Section 4.2. Related asynchronous approaches by Janusevskis et al. [21] and Snoek et al. [34] equally form a first-moment approximation via Monte Carlo sampling and are thus plagued by similar issues.

Hallucination A further approach to dealing with the unobserved function values are hallucination-based methods, such as the Kriging Believer [16]. 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, $\alpha(\cdot)$,

$$\alpha_{KB}(x) = \alpha(x \mid D_n, (\mathcal{B}, \mu_b)), \tag{20}$$

with μ_b as in Equation (11).

While developed for synchronous q-batch construction, the constant liar heuristic offers an additional mechanism to account for locations under evaluation [15].

Thompson sampling Thompson sampling (TS) is an acquisition rule based on sampling the surrogate posterior [37]. 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 from the surrogate posterior. The probability that a particular input being is the true maximizer, x^*, is given by

$$p(x^*|D_n) = \int p(x^* \mid g)\, p(g \mid D_n) dg \tag{21}$$

$$= \int \delta_{\arg\max_{x \in \mathcal{X}} g(x)}(x^*)\, p(g \mid D_n) dg, \tag{22}$$

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. [41], 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. [23]. 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. [23], 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. [2]. This likely explains the inferior performance of standard EI in their work compared to ours.

AEGIS Proposed by De Ath et al. [9], 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

$$\epsilon_T = \epsilon_P = \epsilon/2 \tag{23}$$

$$\epsilon = \min\{2/\sqrt{d}, 1\}, \tag{24}$$

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

De Ath et al. [9] 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 by Alvi et al. [1] 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 [18] (Figure 5). 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. [1] design the local penalizer centered at busy location x_j, $\varphi : \mathcal{X} \mapsto [0, 1]$, as

$$\varphi(x \mid x_j) = \min \left\{ \frac{\hat{L}\|x - x_j\|}{\mid \mu(x_j) - y_n^* \mid + \gamma\sigma(x_j)}, 1 \right\}. \tag{25}$$

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

$$\alpha_{LP}(x \mid \mathcal{B}) = \alpha_{UCB}(x) \prod_{j=1}^{q-1} \varphi(x \mid x_j). \tag{26}$$

Additionally, Alvi et al. [1] 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.

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 Equaiton (3)) does not depend on the as of yet unknown outputs. This makes the second expectation trivial, allowing the objective to be written as

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

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 [16].

D Implementation details

A zero-mean Gaussian Process prior with RBF kernel and ARD is used to form the surrogate in all experiments. The kernel hyperparameters, ϕ, and the observation noise, σ_y^2, were fit to optimize the marginal likelihood of the data [40]. 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 [6]. We employ a multi-restart strategy, where the best 10 from an initial set of $1000d$ candidates are optimized.

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 workers, are drawn from a randomly perturbed Halton sequence in the appropriate dimension [20].

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

We implement our optimization pipeline in BoTorch [4] and GPyTorch [14], and make the code available at https://github.com/Ben-Riegler/AsyncBO_EXAIT. The implementation of AEGIS is adopted from the authors De Ath et al. [9], where the approximate Pareto front is found with code from https://github.com/georgedeath/aegis/blob/main/aegis/batch/nsga2_pygo.py.

E Optimization tasks

E.1 Synthetic test functions

The synthetic test functions we use are available as part of the BoTorch package [4]. 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.

E.2 Real-world tasks

PROTEUS PROTEUS is a coupled atmosphere-interior framework to simulate the temporal evolution of rocky planets [28, 30]. This deterministic forward simulator takes in an 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

$$\min_{x \in \mathcal{X}} \|PROTEUS(x) - \gamma_0\|^2, \tag{27}$$

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. PROETEUS 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 [31]. 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. Following a common approach, we discretize 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.

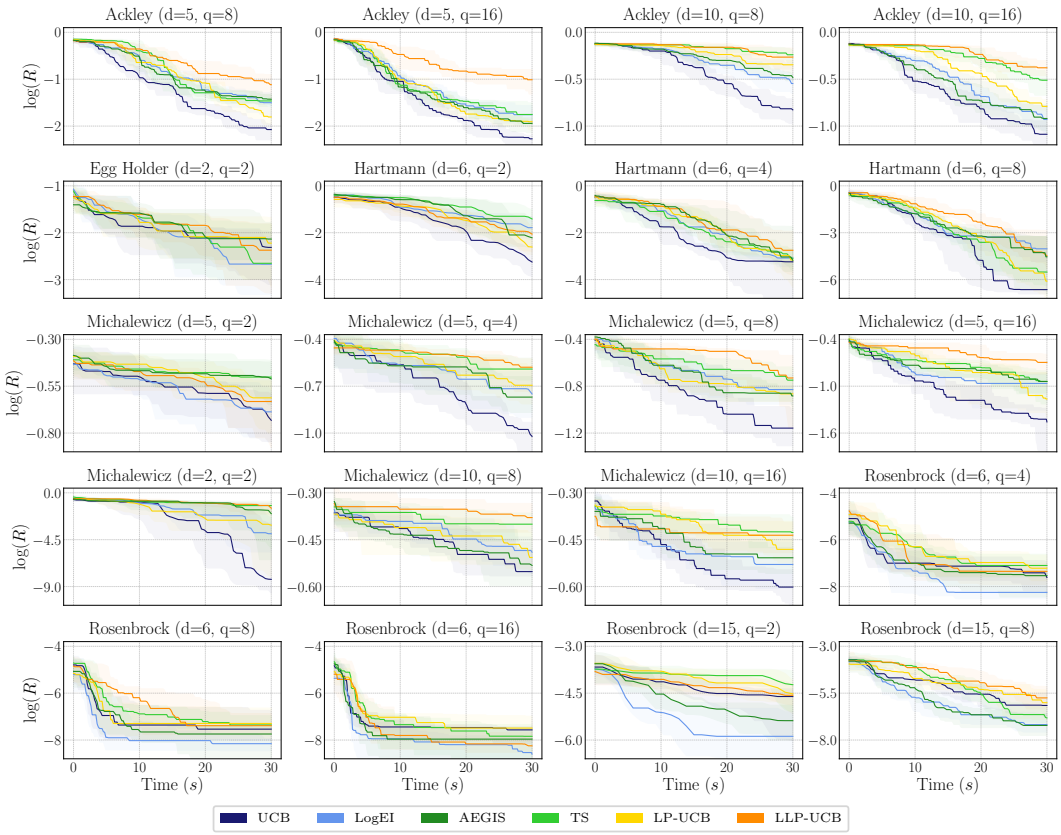

Figure 6: Standard acquisition functions are not significantly outperformed on any of our synthetic test functions. In fact, for our experiments, UCB often outperforms alternatives designed for asynchronous BO.

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 [42]. The hyperparameters optimized are learning rate, number of boosting rounds (trees), maximum tree depth, minimum loss reduction to make split, fraction of training examples to grow 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 [26] contains $60,000$ 32x32 pixel images in ten classes. We randomly select a training and a validation set of $10,000$ 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 $\log(R)$.

F Additional experimental results

In Figures 6 to 9, we give additional experimental results on variants of the problems presented in Section 5.

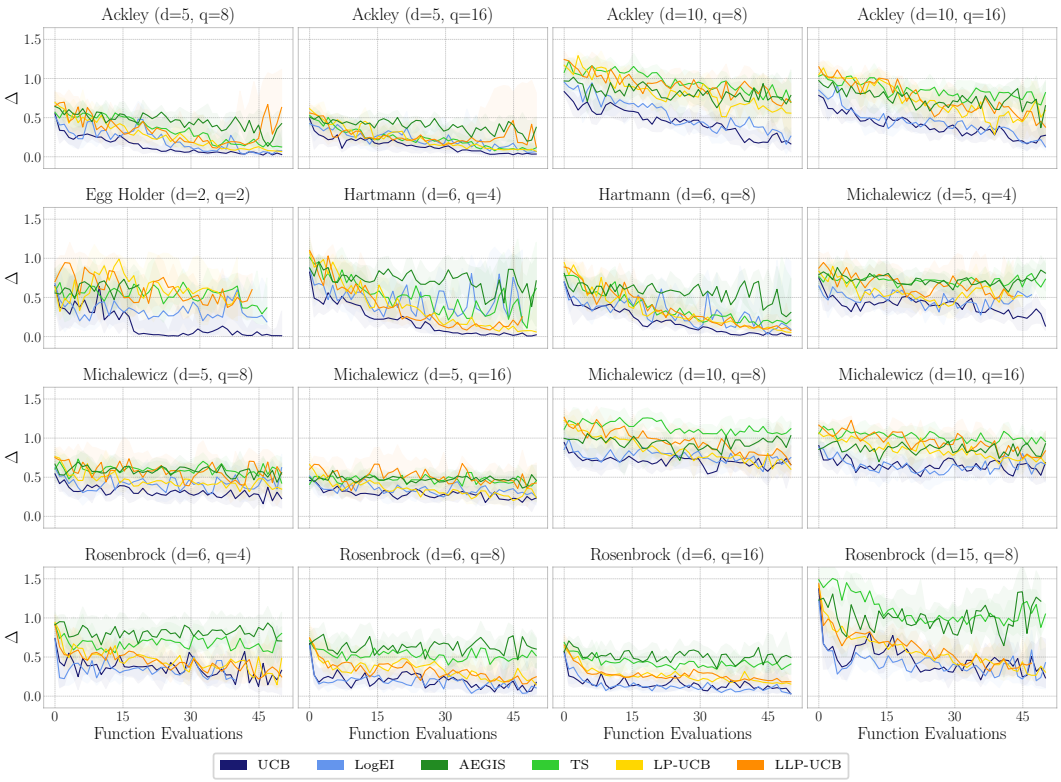

Figure 7: Standard acquisition functions query closer to busy locations than alternatives designed for asynchronous BO, but do not repeat queries.

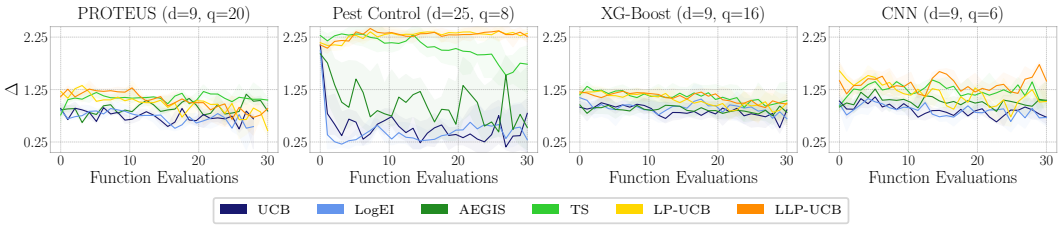

Figure 8: While standard acquisition queries the closest to busy locations, distances on real-world tasks are significantly larger than zero for all methods.

G 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.

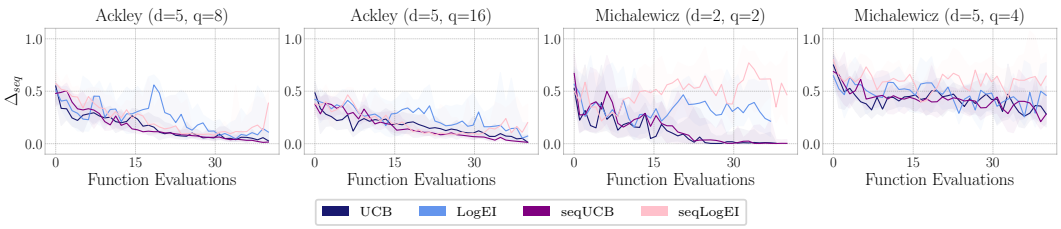

Figure 9: Standard acquisition functions mimic the distance profile of their respective sequential counterparts.