Maximizing acquisition functions for Bayesian optimization

James T. Wilson[*]
Imperial College London

Frank Hutter
University of Freiburg

Marc Peter Deisenroth
Imperial College London
PROWLER.io

Abstract

Bayesian optimization is a sample-efficient approach to global optimization that relies on theoretically motivated value heuristics (acquisition functions) to guide the search process. Fully maximizing acquisition functions produces the Bayes' decision rule, but this ideal is difficult to achieve since these functions are frequently non-trivial to optimize. This statement is especially true when evaluating queries in parallel, where acquisition functions are routinely non-convex, high-dimensional, and intractable. We present two modern approaches for maximizing acquisition functions that exploit key properties thereof, namely the differentiability of Monte Carlo integration and the submodularity of parallel querying.

1 Introduction

Bayesian optimization (BO) is a powerful framework for tackling complicated global optimization problems [30, 38, 43]. Given a black-box function $f : \mathcal{X} \to \mathcal{Y}$, BO seeks to identify a maximizer $\mathbf{x}^* \in \arg\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$ while simultaneously minimizing incurred costs. Recently, these strategies have demonstrated state-of-the-art results on many important, real-world problems ranging from material sciences [15, 56], to robotics [3, 7], to algorithm tuning and configuration [27, 52, 55]

From a high-level perspective, BO can be understood as the application of Bayesian decision theory to optimization problems [10, 13, 44]. One first specifies a belief over possible explanations for f using a probabilistic surrogate model and then combines this belief with an acquisition function $\mathcal{L}$ to convey the expected utility for evaluating a set of queries $\mathbf{X}$. In theory, $\mathbf{X}$ is chosen via Bayes' decision rule as $\mathcal{L}$'s maximizer by solving for an *inner optimization problem* [17, 41, 58]. In practice, challenges associated with maximizing $\mathcal{L}$ greatly impede our ability to live up to this standard. Nevertheless, this inner optimization problem is often treated as a black box unto itself. Failing to address this challenge leads to a systematic departure from BO's fundamentals and, consequently, consistent deterioration in achieved performance.

To help reconcile theory and practice, we present two modern approaches for addressing BO's inner optimization problem that exploit key properties of acquisition functions and their estimators. First, we clarify how sample path derivatives can be used to optimize a broad class of acquisition functions estimated via Monte Carlo (MC) integration. Second, we show that parallel formulations for many of the same acquisition functions are submodular and can therefore be greedily maximized with guaranteed near-optimality. Finally, we demonstrate through comprehensive experiments that these theoretical contributions directly translate to reliable and, often, substantial performance gains.

2 Background

Bayesian optimization relies on both a surrogate model $\mathcal{M}$ and an acquisition function $\mathcal{L}$ to define a strategy for efficiently maximizing a black-box function f. At each "outer-loop" iteration (Fig-

[*]Correspondence to j.wilson17@imperial.ac.uk

Preprint. Work in progress.

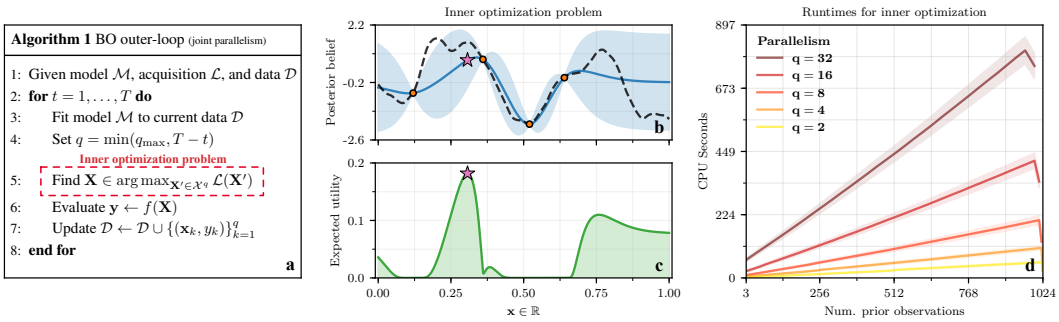

Figure 1: (a) Pseudo-code for standard BO's "outer-loop" with parallelism q; the inner optimization problem is boxed in red. (b)–(c) GP-based belief and expected utility (EI), given four initial observations '•'. The aim of the inner optimization problem is to find the optimal query '☆'. (d): Time to compute 2^{14} evaluations of MC q-EI using a GP surrogate for varied observation counts and degrees of parallelism. Runtimes fall off at the final step because q decreases to accommodate evaluation budget $T = 1,024$.

ure 1a), this strategy is used to choose a set of queries $\mathbf{X}$ whose evaluation advances the optimization process. This section reviews related concepts and closes with discussion of the associated inner optimization problem. For an in-depth BO review, we defer to the recent survey [51]. Without loss of generality, we assume BO strategies evaluate q designs $\mathbf{x} \in \mathbb{R}^d$ in parallel so that setting $q = 1$ recovers purely sequential decision-making. We denote available information regarding f at time t as $\mathcal{D}_t = \{(\mathbf{x}_i, y_i)\}_{i=1}^t$ and any parameters associated with surrogate model $\mathcal{M}$ and acquisition function $\mathcal{L}$ as $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$, respectively. Strictly for notational convenience, we assume noiseless observations $\mathbf{y} = f(\mathbf{X})$. Henceforth, direct reference to the above terms will be omitted where possible.

Surrogate models Surrogate models $\mathcal{M}$ provide a probabilistic interpretation of f whereby possible explanations for the function are seen as draws $f \sim p(f|\mathcal{D})$. In some cases, this belief is expressed as an explicit ensemble of sample functions [26, 53, 59]. More commonly however, $\mathcal{M}$ dictates the parameters $\boldsymbol{\theta}$ of a (joint) distribution over the function's behavior at a finite set of points $\mathbf{X}$. By first tuning the model's (hyper)parameters $\boldsymbol{\phi}$ to explain for $\mathcal{D}$, a belief is formed as $p(\mathbf{y}|\mathbf{X}, \mathcal{D}) = p(\mathbf{y}; \boldsymbol{\theta})$ with $\boldsymbol{\theta} \leftarrow \mathcal{M}(\mathbf{X}; \boldsymbol{\phi})$. Generally, $\boldsymbol{\theta} \leftarrow \mathcal{M}(\mathbf{X}; \boldsymbol{\phi})$ is used to denote that belief p's parameters $\boldsymbol{\theta}$ are specified by model $\mathcal{M}$ evaluated at $\mathbf{X}$. A member of this latter category, the Gaussian process prior (GP) is the most widely used surrogate and induces a multivariate normal belief $\boldsymbol{\theta} \triangleq (\boldsymbol{\mu}, \boldsymbol{\Sigma}) \leftarrow \mathcal{M}(\mathbf{X}; \boldsymbol{\phi})$ such that $p(\mathbf{y}|\mathbf{X}) = \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ for any finite $\mathbf{X}$ (see Figure 1b).

Acquisition functions With few exceptions, acquisition functions amount to integrals defined in terms of a belief p over the unknown outcomes $\mathbf{y} = \{y_1, \ldots, y_q\}$ revealed when evaluating a black-box function f at corresponding input locations $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_q\}$. This formulation naturally occurs as part of a Bayesian approach whereby the value of evaluating $\mathbf{X}$ is determined by accounting for the utility provided by possible outcomes $\mathbf{y} \sim p(\mathbf{y}|\mathbf{X})$. Denoting the chosen utility function as ℓ, this paradigm leads to acquisition functions defined as expectations

$$\mathcal{L}(\mathbf{X}; \mathcal{D}, \boldsymbol{\psi}) = \mathbb{E}_{\mathbf{y}}\left[\ell(\mathbf{y}; \boldsymbol{\psi})\right] = \int \ell(\mathbf{y}; \boldsymbol{\psi}) p(\mathbf{y}|\mathbf{X}, \mathcal{D}) d\mathbf{y}\,. \tag{1}$$

A seeming exception to this rule, *non-myopic* acquisition functions assign value by further considering how different realizations of $\mathcal{D}' \leftarrow \mathcal{D} \cup \{(\mathbf{x}_k, y_k)\}_{k=1}^q$ impact our broader understanding of f and generally correspond to more complex, nested integrals. Figure 1c portrays a prototypical acquisition surface and Table 1 exemplifies popular, myopic and non-myopic instances of (1).

Inner optimization problem Maximizing acquisition functions plays a critical role in BO as the process through which abstract machinery (e.g. model $\mathcal{M}$ and acquisition function $\mathcal{L}$) yields concrete actions (e.g. a decision regarding a set of queries $\mathbf{X}$). Despite its importance however, this inner optimization problem is often neglected. This lack of emphasis is largely attributable to a greater emphasis on creating new and improved machinery as well as applying BO to new types of problems. Moreover, elementary examples of BO facilitate $\mathcal{L}$'s maximization. For example, optimizing a single query $\mathbf{x} \in \mathbb{R}^d$ is usually straightforward when $\mathbf{x}$ is low-dimensional and $\mathcal{L}$ is myopic.

Outside of like rudimentary examples however, BO's inner optimization problem is qualitatively more difficult to solve. In virtually all cases, acquisition functions are non-convex (frequently due to

Abbr.	Acquisition Function $\mathcal{L}$	Reparameterization	$\nabla\mathcal{L}_m$	SM
EI	$\mathbb{E}_{\mathbf{y}}[\max(\text{ReLU}(\mathbf{y}-\alpha))]$	$\mathbb{E}_{\mathbf{z}}[\max(\text{ReLU}(\boldsymbol{\mu}+\mathbf{L}\mathbf{z}-\alpha))]$	Y/Y	Y
PI	$\mathbb{E}_{\mathbf{y}}[\max(\mathbb{1}^{-}(\mathbf{y}-\alpha))]$	$\mathbb{E}_{\mathbf{z}}[\max(\sigma(\frac{\boldsymbol{\mu}+\mathbf{L}\mathbf{z}-\alpha}{\tau}))]$	N/Y	Y
SR	$\mathbb{E}_{\mathbf{y}}[\max(\mathbf{y})]$	$\mathbb{E}_{\mathbf{z}}[\max(\boldsymbol{\mu}+\mathbf{L}\mathbf{z})]$	Y/Y	Y
UCB	$\mathbb{E}_{\mathbf{y}}[\max(\boldsymbol{\mu}+\sqrt{\beta\pi/2}\lvert\boldsymbol{\gamma}\rvert)]$	$\mathbb{E}_{\mathbf{z}}[\max(\boldsymbol{\mu}+\sqrt{\beta\pi/2}\lvert\mathbf{L}\mathbf{z}\rvert)]$	Y/Y	Y
ES	$-\mathbb{E}_{\mathbf{y}_a}[\text{H}(\mathbb{E}_{\mathbf{y}_b\lvert\mathbf{y}_a}[\mathbb{1}^{+}(\mathbf{y}_b-\max(\mathbf{y}_b))])]$	$-\mathbb{E}_{\mathbf{z}_a}[\text{H}(\mathbb{E}_{\mathbf{z}_b}[\text{softmax}(\frac{\boldsymbol{\mu}_{b\lvert a}+\mathbf{L}_{b\lvert a}\mathbf{z}_b}{\tau})])]$	N/Y	N
KG	$\mathbb{E}_{\mathbf{y}_a}[\max(\boldsymbol{\mu}_b+\boldsymbol{\Sigma}_{b,a}\boldsymbol{\Sigma}_{a,a}^{-1}(\mathbf{y}_a-\boldsymbol{\mu}_a))]$	$\mathbb{E}_{\mathbf{z}_a}[\max(\boldsymbol{\mu}_b+\boldsymbol{\Sigma}_{b,a}\boldsymbol{\Sigma}_{a,a}^{-1}\mathbf{L}_a\mathbf{z}_a)]$	Y/Y	N

Table 1: Examples of reparameterizable acquisition functions, the final two column indicate Monte Carlo differentiability and submodularity. Glossary: $\mathbb{1}^{+/-}$ denotes the right-/left-continuous Heaviside step function; ReLU and σ rectified linear and sigmoid nonlinearities, respectively; H the Shannon entropy; α an improvement threshold; τ a temperature parameter; $\mathbf{L}\mathbf{L}^\top \triangleq \boldsymbol{\Sigma}$ the Cholesky factor; and, residuals $\boldsymbol{\gamma} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. Lastly, non-myopic acquisition function (ES and KG) are assumed to be defined using a discretization. Terms associated with the query set and discretization are respectively denoted using subscripts a and b.

the non-convexity of plausible explanations for f). Accordingly, increases in input dimensionality d can be prohibitive to efficient query optimization. In the generalized setting with parallelism $q \geq 1$, this issue is exacerbated by additional scaling in q. To the extent that this combination of non-convexity and (acquisition) dimensionality is problematic though, the routine intractability of both non-myopic and parallel acquisition poses a commensurate challenge.

As is generally true of integrals, the majority of acquisition functions are intractable. Even Gaussian integrals, which are often preferred because they lead to analytic solutions for certain instances of (1), are only tractable in a handful of special cases [12, 16, 18]. To circumvent the lack of closed-form solutions, researchers have proposed a diversity of methods. Approximation strategies [12, 14, 59], which replace a quantity of interest with a more readily computable one, work well in practice but may not to converge to the true value. In contrast, bespoke solutions [9, 18, 20] provide (near-)analytic expressions but typically do not scale well with dimensionality.[2] Lastly, MC methods [25, 46, 52] are highly versatile and usually unbiased but are often perceived as non-differentiable and, therefore, inefficient for purposes of maximizing $\mathcal{L}$.

Regardless of the method however, the (often drastic) increase in cost when evaluating $\mathcal{L}$'s proxy acts as a barrier to efficient query optimization, and these costs increase over time as shown in Figure 1d. In an effort to address these problems, we now go inside the outer-loop and focus on efficient methods for maximizing acquisition functions.

3 Maximizing acquisition functions

This section presents the technical contributions of this paper, which can be broken down into two complementary topics: 1) gradient-based optimization of acquisition functions that are estimated via Monte Carlo integration, and 2) submodular maximization of parallel acquisition functions. Below, we separately discuss each contribution along with its related literature.

3.1 Differentiability of Monte Carlo acquisitions

Gradients are one of the most valuable sources of information for optimizing a function. In this section, we detail both the reasons and conditions whereby MC acquisition functions are differentiable and further show that most well-known examples satisfy these criteria (see Table 1).

We assume that $\mathcal{L}$ is an expectation over a multivariate normal belief $p(\mathbf{y}|\mathbf{X}) = \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ specified by a GP surrogate such that $(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \leftarrow \mathcal{M}(\mathbf{X})$. More generally, we assume that samples can be generated as $\mathbf{y}_k \sim p(\mathbf{y}|\mathbf{X})$ to form an unbiased MC estimator of an acquisition function as $\mathcal{L}(\mathbf{X}) \approx \mathcal{L}_m(\mathbf{X}) \triangleq \frac{1}{m}\sum_{k=1}^{m} \ell(\mathbf{y}_k)$. Given such an estimator, we are interested in verifying whether

$$\nabla\mathcal{L}(\mathbf{X}) \approx \nabla\mathcal{L}_m(\mathbf{X}) \triangleq \frac{1}{m}\sum_{k=1}^{m} \nabla\ell(\mathbf{y}_k), \tag{2}$$

[2] By *near-analytic*, we refer to cases where an expression contains terms that cannot be computed exactly but for which high-quality solvers exist (e.g. low-dimensional multivariate normal CDF estimators [18, 19]).

where $\nabla\ell(\mathbf{y}_k)$ denotes the gradient of utility function ℓ taken with respect to $\mathbf{X}$. The validity of MC gradient estimator (2) is obscured by the fact that sample $\mathbf{y}_k$ depends on $\mathbf{X}$ through generative distribution p and that $\nabla\mathcal{L}_m$ is no longer the derivative of ℓ's expectation.

Originally referred to as *infinitesimal perturbation analysis* [8, 22], the *reparameterization trick* [35, 49] is the process of differentiating through an MC estimate to its generative distribution p's parameters and consists of two components: i) reparameterizing samples from p as draws from a simpler base distribution $\hat{p}$, and ii) interchanging differentiation and integration by taking the expectation over sample path derivatives.

Reparameterization Reparameterization is a way of interpreting samples that makes their differentiability w.r.t. a generative distribution's parameters transparent. Often, samples $\mathbf{y} \sim p(\mathbf{y}; \boldsymbol{\theta})$ can be re-expressed as a deterministic mapping $\pi : \mathcal{Z} \times \boldsymbol{\Theta} \to \mathcal{Y}$ of simpler random variables $\mathbf{z} \sim \hat{p}(\mathbf{z})$ [35]. This change of variables helps clarify that, if ℓ is a differentiable function of $\mathbf{y} = \pi(\mathbf{z}; \boldsymbol{\theta})$, then $\frac{d\ell}{d\boldsymbol{\theta}} = \frac{d\ell}{d\pi}\frac{d\pi}{d\boldsymbol{\theta}}$ by the chain rule of (functional) derivatives.

If generative distribution p is multivariate normal with parameters $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the corresponding mapping is then $\pi(\mathbf{z}; \boldsymbol{\theta}) \triangleq \boldsymbol{\mu} + \mathbf{L}\mathbf{z}$, where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\mathbf{L}$ is $\boldsymbol{\Sigma}$'s Cholesky factor s.t. $\mathbf{L}\mathbf{L}^\top = \boldsymbol{\Sigma}$. Rewriting (1) as a Gaussian integral and reparameterizing, we have

$$\mathcal{L}(\mathbf{X}) = \int_{\boldsymbol{a}}^{\boldsymbol{b}} \ell(\mathbf{y})\mathcal{N}(\mathbf{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma})d\mathbf{y} = \int_{\boldsymbol{a}'}^{\boldsymbol{b}'} \ell(\boldsymbol{\mu} + \mathbf{L}\mathbf{z})\mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})d\mathbf{z}\,, \tag{3}$$

where each of the q terms c_i' in both $\boldsymbol{a}'$ and $\boldsymbol{b}'$ is transformed as $c_i' = (c_i - \mu_i - \sum_{j<i} L_{ij}z_j)/L_{ii}$. The third column of Table 1 grounds (3) with several prominent examples. For a given draw $\mathbf{y}_k \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the sample path derivate of ℓ w.r.t. $\mathbf{X}$ is therefore

$$\nabla\ell(\mathbf{y}_k) = \frac{d\ell(\mathbf{y}_k)}{d\mathbf{y}_k}\frac{d\mathbf{y}_k}{d\mathcal{M}(\mathbf{X})}\frac{d\mathcal{M}(\mathbf{X})}{d\mathbf{X}}, \tag{4}$$

where, by minor abuse of notation, we have substituted in $\mathbf{y}_k = \pi\left(\mathbf{z}_k; \mathcal{M}(\mathbf{X})\right)$. Hence, reinterpreting $\mathbf{y}$ as a function of $\mathbf{z}$ sheds light on individual MC sample's differentiability.

Interchangeability Since $\mathcal{L}_m$ is an unbiased MC estimator consisting of differentiable terms, it is natural to wonder whether the average sample gradient $\nabla\mathcal{L}_m$ (2) follows suit, i.e., whether

$$\nabla\mathcal{L}(\mathbf{X}) = \nabla\mathbb{E}_{\mathbf{y}}\left[\ell(\mathbf{y})\right] \overset{?}{=} \mathbb{E}_{\mathbf{y}}\left[\nabla\ell(\mathbf{y})\right] \approx \nabla\mathcal{L}_m(\mathbf{X})\,, \tag{5}$$

where $\overset{?}{=}$ denotes the potential equivalence when interchanging differentiation and expectation. Necessary and sufficient conditions for this interchange are that, as defined under p, integrand ℓ must be continuous and its first derivative ℓ' must a.s. exist and be integrable [8, 22]. Independent of but prior to our work, Wang et al. [58] demonstrated that these conditions are met for a GP with a twice differentiable kernel, provided that the elements in query set $\mathbf{X}$ are unique. The authors then use these results to prove that (2) is an unbiased gradient estimator for the parallel Expected Improvement (q-EI) acquisition function [9, 20, 52]. In later works, these findings were extended to include the parallel Knowledge Gradient (KG) acquisition function [61]. Figure 2d (bottom right) visualizes gradient-based optimization of MC q-EI for parallelism $q = 2$.

Extensions Rather than focusing on individual examples, our goal is to show differentiability for a broad class of MC acquisition functions. In addition to its conceptual simplicity, one of MC integration's primary strengths is its generality. This versatility is evident in Table 1, which catalogs (differentiable) reparameterizations for six of the most popular acquisition functions. While some of these forms were previously known (EI and KG) or follow freely from the above (SR), others require additional steps. We summarize these steps below and provide full details in Appendix A.

In many cases of interest, utility is measured in terms of discrete events. For example, Probability of Improvement [38, 57] is the expectation of a binary event e_{PI}: "will a new set of results improve over the preceding ones?". Similarly, Entropy Search [25] contains expectations of categorical events e_{ES}: "which of a set of random variables will be the largest?" Unfortunately, mappings from continuous samples $\mathbf{y}$ to discrete events e are typically discontinuous and, therefore, violate the conditions for (5). To overcome this issue, we utilize *concrete* (continuous to discrete) approximations in place of the original, discontinuous mappings.

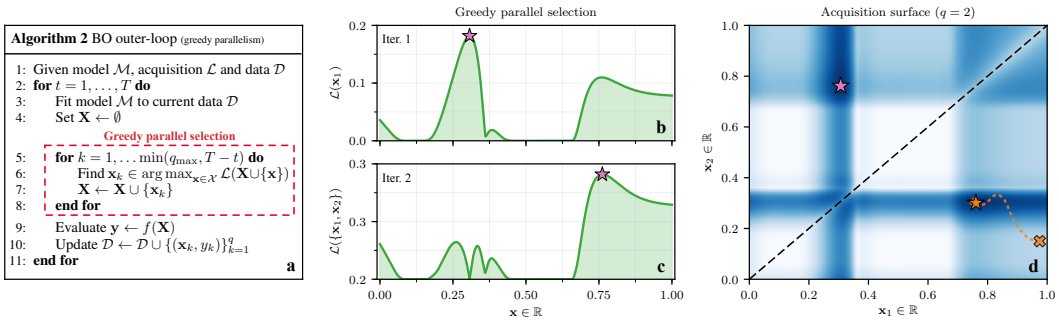

Figure 2: a) Pseudo-code for BO outer-loop with greedy parallelism, the inner optimization problem is boxed in red. b-c) Successive iterations of greedy maximization, starting from the posterior shown in Figure 1b. d) On the left, greedily selected query '☆'; on the right and from '×' to '☆', trajectory when jointly optimizing parallel queries $\mathbf{x}_1$ and $\mathbf{x}_2$ via stochastic gradient ascent. Darker colors correspond with larger acquisitions.

Still within the context of the reparameterization trick, [29, 40] studied the closely related problem of optimizing an expectation w.r.t. a discrete generative distribution's parameters. To do so, the authors propose relaxing the mapping from, e.g., uniform to categorical random variables with a continuous approximation so that the (now differentiable) transformed variables closely resemble their discrete counterparts in distribution. In our case, we first map from uniform to Gaussian random variables, but the general process is otherwise identical. Concretely, we can approximate PI's binary event as

$$\tilde{e}_{\text{PI}}(\mathbf{X}; \alpha, \tau) = \max\left(\sigma\left({}^{\mathbf{y}-\alpha}/_{\tau}\right)\right) \approx \max\left(\mathbb{1}^{-}(\mathbf{y} - \alpha)\right) \tag{6}$$

where $\mathbb{1}^{-}$ denotes the left-continuous Heaviside step function, σ the sigmoid nonlinearity, and $\tau \in [0, \infty]$ acts as a temperature parameter s.t. the approximation becomes exact as $\tau \to 0$. Appendix A.1 further discusses concrete approximations for both PI and ES.

Lastly, the Upper Confidence Bound (UCB) acquisition function [54] is typically not portrayed as an expectation, seemingly barring the use of MC methods. At the same time, the standard definition $\text{UCB}(\mathbf{x}; \beta) \triangleq \mu + \beta^{1/2}\sigma$ bares a striking resemblance to the reparameterization for normal random variables $\pi(z; \mu, \sigma) = \mu + \sigma z$. By exploiting this insight, it is possible to rewrite this closed-form expression as $\text{UCB}(\mathbf{x}; \beta) = \int_{\mu}^{\infty} y\mathcal{N}(y; \mu, 2\pi\beta\sigma^2)dy$. Formulating UCB as an expectation allows us to naturally parallelize this acquisition function as

$$\text{UCB}(\mathbf{X}; \beta) = \mathbb{E}_{\mathbf{y}}\left[\max(\boldsymbol{\mu} + \sqrt{{}^{\beta\pi}/_{2}}|\boldsymbol{\gamma}|)\right], \tag{7}$$

where $\boldsymbol{\gamma} = \mathbf{y} - \boldsymbol{\mu}$ denotes $\mathbf{y}$'s residuals. In contrast with existing parallelizations of UCB [11, 14], Equation (7) directly generalizes its marginal form and can be efficiently estimated via MC integration (see Appendix A.2 for the full derivation).

3.2 Submodularity of parallel acquisitions

Commonly associated with the notion of *diminishing returns*, submodularity is an important property of set functions with far-reaching theoretical and practical implications [2, 37]. Established results [37, 42, 45] provide regret bounds for greedy approaches to maximizing submodular functions. Specifically, if $\mathcal{L}$ is a normalized submodular function with maximum $\mathcal{L}^*$, then a greedy maximizer is guaranteed to incur at most $\frac{1}{e}\mathcal{L}^*$ regret.

In the context of BO, submodularity has previously been appealed to when establishing outer-loop regret bounds [11, 14, 54]. Like applications of submodularity utilize this property by relating an idealized BO strategy to greedy maximization of a submodular objective (e.g. the mutual information between unknown function f and observations $\mathcal{D}$). In contrast, we directly show submodularity for various parallel acquisitions functions per se. Since non-negative linear combinations of submodular functions are submodular, it suffices to show that the integrand ℓ of an expected utility function (1) is submodular.

Submodularity of the max In Table 1, the $\max$ operator acts as the common thread tying together many parallel acquisition functions. Accordingly, we begin by showing that the $\max$ is indeed submodular. Let $\max(\emptyset) = \min(\mathcal{Y}) = c$, then given sets $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{Y}$ and $\forall y \in \mathcal{Y}$,

$$\max(\mathcal{A} \cup \{y\}) - \max(\mathcal{A}) \geq \max(\mathcal{B} \cup \{y\}) - \max(\mathcal{B}). \tag{8}$$

This assertion can be proven using the equivalent definition: $\max(\mathcal{A}) + \max(\mathcal{B}) \geq \max(\mathcal{A} \cup \mathcal{B}) + \max(\mathcal{A} \cap \mathcal{B})$. Without loss of generality, assume $\max(\mathcal{A} \cup \mathcal{B}) = \max(\mathcal{A})$; then, $\max(\mathcal{B}) \geq \max(\mathcal{A} \cap \mathcal{B})$ since, for any $\mathcal{C} \subseteq \mathcal{B}$, $\max(\mathcal{B}) \geq \max(\mathcal{C})$. Guarantees for submodular maximization pertain to normalized set functions though, so c must be bounded from below such that the $\max$ can be redefined with $\max(\emptyset) = 0$.

Normalizing utility functions Lower bounding $c > -\infty$ is facilitated by the fact that this term pertains to the arguments of the $\max$ rather than to (a belief over) black-box f. However, because the $\max$ operator is additive such that $\max(\mathbf{y} - c) = \max(\mathbf{y}) - c$, the following normalizations are only necessary when establishing regret bounds. Addressing the matter by case, we have:

a. Expected Improvement: For a given threshold α, let improvement be defined (element-wise) as $\mathrm{ReLU}(\mathbf{y} - \alpha) = \max(0, \mathbf{y} - \alpha)$. EI's integrand is then the largest improvement $\ell_{\mathrm{EI}}(\mathbf{y}; \alpha) = \max(\mathrm{ReLU}(\mathbf{y} - \alpha))$.[3] Applying the rectifier prior to the $\max$ defines ℓ_{EI} as a normalized, submodular function.

b. Probability of Improvement: PI's integrand is defined as $\ell_{\mathrm{PI}}(\mathbf{y}, \alpha) = \max(\mathbb{1}^{-}(\mathbf{y} - \alpha))$, where $\mathbb{1}^{-}$ denotes the left-continuous Heaviside step function. Seeing as the Heaviside maps $\mathcal{Y} \mapsto \{0, 1\}$, ℓ_{PI} is already normalized.

c. Simple Regret: The submodularity of Simple Regret was previously discussed in [1], under the assumption $c = 0$. More generally, normalizing ℓ_{SR} requires bounding f's infimum under p. Technical challenges for doing so make submodular maximization of SR the hardest to justify.

d. Upper Confidence Bound: As per (7), define UCB's integrand as the maximum over $\mathbf{y}$'s expectation incremented by non-negative terms. By definition then, ℓ_{UCB} is lower bounded by the predictive mean and can therefore be normalized as $\bar{\ell}_{\mathrm{UCB}} = \max(\boldsymbol{\mu} + |\boldsymbol{\gamma}| - c)$, provided that $c = \min_{\mathbf{x} \in \mathcal{X}} \mu(\mathbf{x})$ is finite. For a zero-mean GP with a twice differentiable kernel, this condition is guaranteed for bounded functions f.

This demonstration of submodularity motivates a greedy approach to the inner optimization problem. Whereas most BO papers discussing greedy parallelism focus on proposing new acquisition functions [1, 11, 14, 33, 60], our results center on established cases. Aside from this distinction however, the submodular maximization procedure proposed here is the same. At each iteration $k = 1, \ldots, q$, we treat the $k-1$ preceding choices $\mathbf{X}$ as constants and grow the set by selecting an additional query $\mathbf{x}_k \in \arg\max_{\mathbf{x} \in \mathcal{X}} \mathcal{L}(\mathbf{X} \cup \{\mathbf{x}\})$. Algorithm 2 in Figure 2 outlines this process as it occurs within the context of the outer-loop. For q-EI, we recover the parallel asynchronous strategy proposed by [21, 52], further justifying its use as a synchronous one [39, 50].

4 Experiments

We assessed the efficacy of gradient-based and submodular strategies for maximizing acquisition function in two primary settings: "synthetic", where task f was drawn from a known GP prior, and "black-box", where f's nature is unknown to the optimizer. In both cases, we used a GP surrogate with a constant mean and an anisotropic Matérn-5/2 kernel. For black-box tasks, ambiguity regarding the correct function prior was handled by fitting GP (hyper)parameters online. Appendix B.1 provides additional details regarding synthetic tasks.

We present results averaged over 32 independent trials. Each trial began with three randomly chosen inputs, and competing methods were run from identical starting conditions. While the general notation of the paper has assumed noise-free f, all experiments were run with Gaussian measurement noise leading to observed values $\hat{y} \sim \mathcal{N}(f(\mathbf{x}), 1\mathrm{e}{-3})$.

Acquisition functions We focused on parallel MC acquisition functions $\mathcal{L}_m$, particularly EI and UCB. Results using EI are shown here and those using UCB are provided in Appendix B.3. In additional experiments, we observed that optimization of PI and SR behaved like that of EI and UCB, respectively. However, overall performance using these acquisition functions was slightly worse, so further results are not reported here. Across experiments, the q-UCB acquisition function introduced in Section 3.1 outperformed q-EI on all tasks but the Levy function.

[3] ℓ_{EI} is often written as the improvement of $\max(\mathbf{y})$; however, these two forms are equivalent.

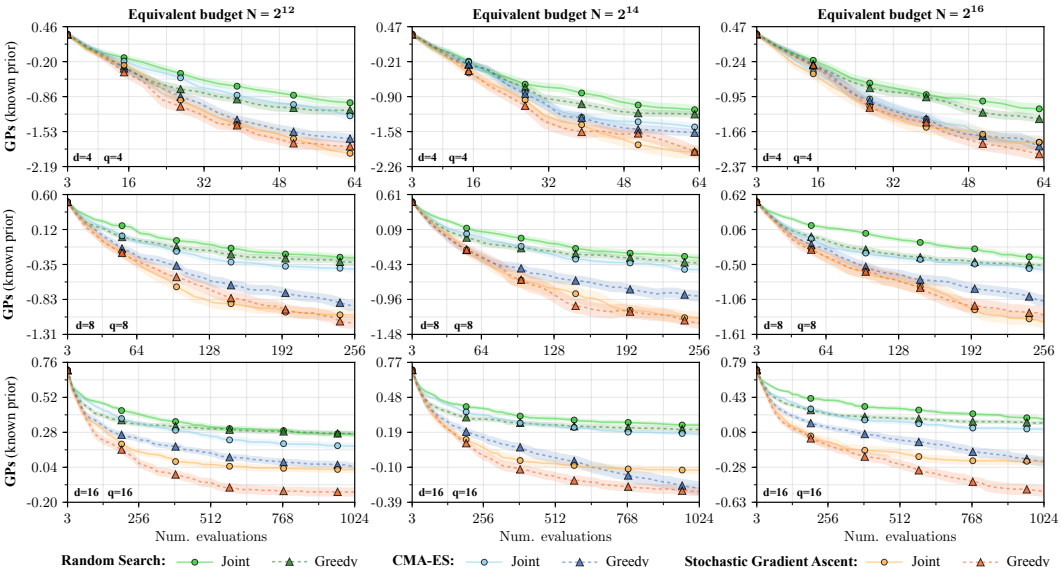

Figure 3: Average performance of different acquisition maximizers on synthetic tasks from a known prior, given varied runtimes when maximizing Monte Carlo q-EI. Reported values indicate the log of the immediate regret $\log_{10} |f_{\max} - f(\mathbf{x}^*)|$, where $\mathbf{x}^*$ denotes the observed maximizer $\mathbf{x}^* \in \arg\max_{\mathbf{x} \in \mathcal{D}} \hat{y}$.

Generally speaking, MC estimators $\mathcal{L}_m$ come in both deterministic and stochastic varieties. Here, determinism refers to whether or not each of m samples $\mathbf{y}_k$ were generated using the same random variates $\mathbf{z}_k$ within a given outer-loop iteration (see Section 3.1). Together with a decision regarding "batch-size" m, this choice reflects a well-known tradeoff of approximation-, estimation-, and optimization-based sources of error when maximizing the true function $\mathcal{L}$ [6]. We explored this tradeoff for each maximizer and summarize our findings below.

Maximizers We considered a range of (acquisition) maximizers, ultimately settling on stochastic gradient ascent (ADAM, [34]), Covariance Matrix Adaptation Evolution Strategy (CMA-ES, [24]) and Random Search (RS, [4]).[4] For fair comparison, maximizers were constrained by CPU runtime. At each outer-loop iteration, an "inner budget" was defined as the average time taken to simultaneously evaluate N acquisition values given equivalent conditions. When using greedy parallelism, this budget was split evenly among each of q iterations. To characterize performance as a function of allocated runtime, experiments were run using inner budgets $N \in \{2^{12}, 2^{14}, 2^{16}\}$.

For ADAM, we used stochastic minibatches consisting of $m = 128$ samples and an initial learning rate $\eta = 1/40$. To combat non-convexity, gradient ascent was run from a total of 32 (64) starting positions when greedily (jointly) maximizing $\mathcal{L}$. Appendix B.2 details the multi-start initialization strategy. As with the gradient-based approaches, CMA-ES performed better when run using stochastic minibatches ($m = 128$). Furthermore, reusing the aforementioned initialization strategy to generate CMA-ES's initial population of 64 samples led to additional performance gains.

Empirical results Figures 3 and 4 present key results regarding BO performance under varying conditions. Both sets of experiments explored an array of input dimensionalities d and degrees of parallelism q (shown in the lower left corner of each panel). Maximizers are grouped by color, with darker colors denoting use of greedy parallelism; inner budgets are shown in ascending order from left to right.

[4] We compared several gradient-based approaches (incl. L-BFGS-B and Polyak averaging [58]) and found that ADAM consistently delivered superior performance. CMA-ES was included after repeatedly outperforming the rival black-box method DIRECT [31]. RS was chosen as a naïve acquisition maximizer after Successive Halving (SH, [28, 32]) failed to yield significant improvement. For consistency when identifying the best proposed query set(s), both RS and SH used deterministic estimators $\mathcal{L}_m$. Whereas RS was run with a constant batch-size $m = 1024$, SH started small and iteratively increased m to refine estimated acquisition values for promising candidates using a cumulative moving average and cached posteriors.

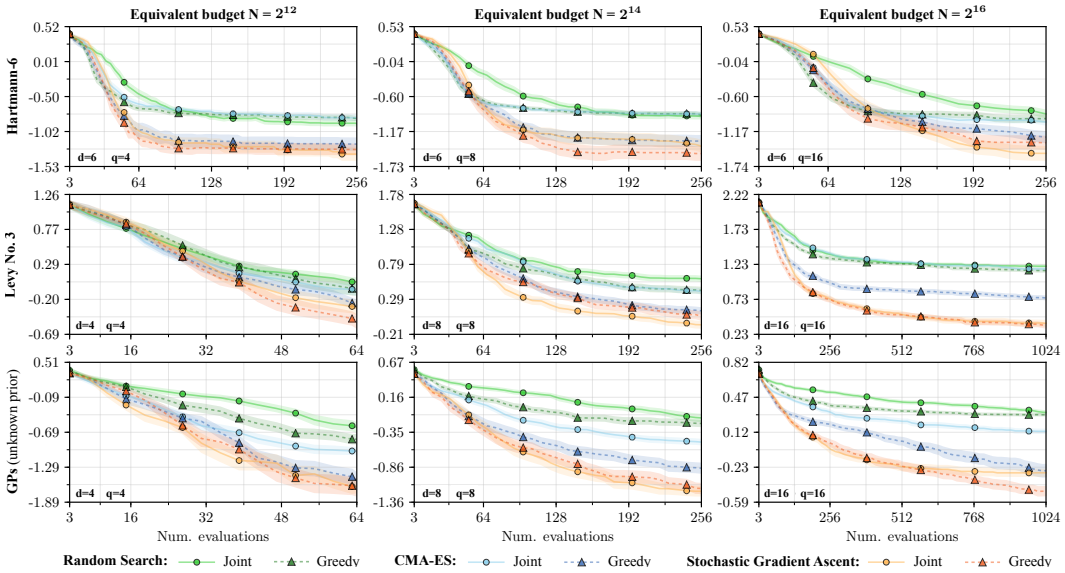

Figure 4: Average performance of different acquisition maximizers on black-box tasks from an unknown prior, given varied runtimes when maximizing Monte Carlo q-EI. Reported values indicate the log of the immediate regret $\log_{10} |f_{\max} - f(\mathbf{x}^*)|$, where $\mathbf{x}^*$ denotes the observed maximizer $\mathbf{x}^* \in \arg\max_{\mathbf{x} \in \mathcal{D}} \hat{y}$.

Results on synthetic tasks (Figure 3), provide a cleaner picture regarding the maximizers' impact on the full BO loop by eliminating the model mismatch. Across all dimensions d (rows) and inner budgets N (columns), gradient-based maximizers (orange) were consistently superior to both gradient-free (blue) and naïve (green) alternatives. Similarly, submodular maximizers generally surpassed their joint counterparts. However, in lower-dimensional cases where gradients alone suffice to optimize $\mathcal{L}_m$, the benefits for coupling gradient-based strategies with near-optima seeking submodular maximization naturally decline. Lastly, the benefits of exploiting gradients and submodularity both scaled with increasing acquisition dimensionality $q \times d$.

Trends are largely identical for black-box tasks (Figure 4), and this commonality is most evident for tasks sampled from an unknown GP prior (final row). These runs were identical to ones on synthetic tasks (specifically, the diagonal of Figure 3) but where knowledge of f's prior was withheld. Outcomes here clarify the impact of model mismatch, showing how maximizers maintain their influence. Finally, performance on Hartmann-6 (top row) serves as a clear indicator of the importance for throughly solving the inner optimization problem. In these experiments, performance improved despite mounting parallelism due to a corresponding increase in the inner budget.

Overall, these results clearly demonstrate that both gradient-based and submodular approaches to (parallel) query optimization lead to reliable and, often, substantial improvement in outer-loop performance. Furthermore, these gains become more pronounced as the acquisition dimensionality increases. Viewed in isolation, maximizers utilizing gradients consistently outperform gradient-free alternatives. Similarly, greedy strategies improve upon their joint counterparts in most cases.

5 Conclusion

Bayesian optimization relies upon an array of powerful tools, such as surrogate models and acquisition functions, and all of these tools are sharpened by strong usage practices. We extend these practices by demonstrating that Monte Carlo acquisition functions provide unbiased gradient estimates that can be exploited when optimizing them. Furthermore, we show that many of the same acquisition functions are submodular when querying in parallel. Comprehensive empirical evidence concludes that both insights lead to substantial performance gains in real-world scenarios where queries must be chosen in finite time. By more efficiently solving for the associated inner optimization problem, our contributions therefore benefit both the theory and practice of Bayesian optimization.

Acknowledgments

The authors would like to thank David Ginsbourger, Dario Azzimonti and Henry Wynn for initial discussions regarding the submodularity of various integrals. The support of the EPSRC Centre for Doctoral Training in High Performance Embedded and Distributed Systems (reference EP/L016796/1) is gratefully acknowledged. This work has partly been supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant no. 716721.

References

[1] J. Azimi, A. Fern, and X.Z. Fern. Batch Bayesian optimization via simulation matching. In *Advances in Neural Information Processing Systems*, 2010.

[2] F. Bach. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends® in Machine Learning*, 6(2-3), 2013.

[3] S. Bansal, R. Calandra, T. Xiao, S. Levine, and C.J. Tomlin. Goal-driven dynamics learning via Bayesian optimization. *arXiv preprint arXiv:1703.09260*, 2017.

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

[5] S. Bochner. *Lectures on Fourier Integrals*. Number 42. Princeton University Press, 1959.

[6] O. Bousquet and L. Bottou. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems*, 2008.

[7] R. Calandra, A. Seyfarth, J. Peters, and M.P. Deisenroth. Bayesian optimization for learning gaits under uncertainty. *Annals of Mathematics and Artificial Intelligence*, 76(1-2), 2016.

[8] X. Cao. Convergence of parameter sensitivity estimates in a stochastic experiment. *IEEE Transactions on Automatic Control*, 30(9), 1985.

[9] C. Chevalier and D. Ginsbourger. Fast computation of the multi-points expected improvement with applications in batch selection. In *International Conference on Learning and Intelligent Optimization*, 2013.

[10] R. Christian. *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer Science & Business Media, 2007.

[11] E. Contal, D. Buffoni, A. Robicquet, and N. Vayatis. Parallel Gaussian process optimization with upper confidence bound and pure exploration. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2013.

[12] J.P. Cunningham, P. Hennig, and S. Lacoste-Julien. Gaussian probabilities and expectation propagation. *arXiv preprint arXiv:1111.6832*, 2011.

[13] M.H. DeGroot. *Optimal statistical decisions*, volume 82. John Wiley & Sons, 2005.

[14] T. Desautels, A. Krause, and J.W. Burdick. Parallelizing exploration-exploitation tradeoffs in Gaussian process bandit optimization. *Journal of Machine Learning Research*, 2014.

[15] P.I. Frazier and J. Wang. Bayesian optimization for materials design. In *Information Science for Materials Discovery and Design*. 2016.

[16] H.I. Gassmann, I. Deák, and T. Szántai. Computing multivariate normal probabilities: A new look. *Journal of Computational and Graphical Statistics*, 11(4), 2002.

[17] M.A. Gelbart, J. Snoek, and R.P. Adams. Bayesian optimization with unknown constraints. *arXiv preprint arXiv:1403.5607*, 2014.

[18] A. Genz. Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 1992.

[19] A. Genz. Numerical computation of rectangular bivariate and trivariate normal and t probabilities. *Statistics and Computing*, 14(3), 2004.

[20] D. Ginsbourger, R. Le Riche, and L. Carraro. *Kriging is well-suited to parallelize optimization*, chapter 6. Springer, 2010.

[21] D. Ginsbourger, J. Janusevskis, and R. Le Riche. Dealing with asynchronicity in parallel Gaussian process based global optimization. In *International Conference of the ERCIM WG on Computing & Statistics*, 2011.

[22] P. Glasserman. Performance continuity and differentiability in Monte Carlo optimization. In *Simulation Conference Proceedings, 1988 Winter*. IEEE, 1988.

[23] I.S. Gradshteyn and I.M. Ryzhik. *Table of integrals, series, and products*. Academic press, 2014.

[24] N. Hansen. The CMA evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2016.

[25] P. Hennig and C. Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 2012.

[26] J. Hernández-Lobato, M. Hoffman, and Z. Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems*, 2014.

[27] F. Hutter, H.H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *International Conference on Learning and Intelligent Optimization*. Springer, 2011.

[28] K. Jamieson and A. Talwalkar. Non-stochastic best arm identification and hyperparameter optimization. In *Artificial Intelligence and Statistics*, 2016.

[29] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with Gumbel-Softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[30] D. Jones, M. Schonlau, and W. Welch. Efficient global optimization of expensive black box functions. *Journal of Global Optimization*, 13:455–492, 1998.

[31] D.R. Jones, C.D. Perttunen, and B.E. Stuckman. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 1993.

[32] Z. Karnin, T. Koren, and O. Somekh. Almost optimal exploration in multi-armed bandits. In *International Conference on Machine Learning*, 2013.

[33] T. Kathuria, A. Deshpande, and P. Kohli. Batched Gaussian process bandit optimization via determinantal point processes. In *Advances in Neural Information Processing Systems*, 2016.

[34] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[35] D.P. Kingma and M. Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

[36] S. Kotz and S. Nadarajah. *Multivariate t-distributions and their applications*. Cambridge University Press, 2004.

[37] A. Krause and D. Golovin. Submodular function maximization, 2014.

[38] H.J. Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1), 1964.

[39] B. Letham, B. Karrer, G. Ottoni, and E. Bakshy. Constrained Bayesian optimization with noisy experiments. *arXiv preprint arXiv:1706.07094*, 2017.

[40] C.J. Maddison, A. Mnih, and Y.W. Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[41] R. Martinez-Cantin. Bayesopt: A Bayesian optimization library for nonlinear optimization, experimental design and bandits. *Journal of Machine Learning Research*, 15(1), 2014.

[42] M. Minoux. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization Techniques*. 1978.

[43] J. Močkus. On Bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference*. Springer, 1975.

[44] J. Močkus. Application of Bayesian approach to numerical methods of global and stochastic optimization. *Journal of Global Optimization*, 4(4), 1994.

[45] G.L. Nemhauser, L.A. Wolsey, and M.L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1), 1978.

[46] M.A. Osborne, R. Garnett, and S.J. Roberts. Gaussian processes for global optimization. In *International Conference on Learning and Intelligent Optimization*, 2009.

[47] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, 2008.

[48] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[49] D.J. Rezende, M. Shakir, and D. Wierstra. Stochastic backpropagation and variational inference in deep latent Gaussian models. In *International Conference on Machine Learning*, 2014.

[50] A. Shah and Z. Ghahramani. Parallel predictive entropy search for batch global optimization of expensive objective functions. In *Advances in Neural Information Processing Systems*, 2015.

[51] B. Shahriari, K. Swersky, Z. Wang, R.P. Adams, and N. de Freitas. Taking the human out of the loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, (1), 2016.

[52] J. Snoek, H. Larochelle, and R.P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25*, 2012.

[53] J.T. Springenberg, A. Klein, S. Falkner, and F. Hutter. Bayesian optimization with robust Bayesian neural networks. In *Advances in Neural Information Processing Systems*, 2016.

[54] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning*, 2010.

[55] K. Swersky, J. Snoek, and R.P. Adams. Multi-task Bayesian optimization. In *Advances in Neural Information Processing Systems*, 2013.

[56] T. Ueno, T.D. Rhone, Z. Hou, T. Mizoguchi, and K. Tsuda. Combo: An efficient Bayesian optimization library for materials science. *Materials discovery*, 4, 2016.

[57] F. Viana and R. Haftka. Surrogate-based optimization with parallel simulations using the probability of improvement. In *AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, 2010.

[58] J. Wang, S.C. Clark, E. Liu, and P.I. Frazier. Parallel Bayesian global optimization of expensive functions. *arXiv preprint arXiv:1602.05149*, 2016.

[59] Z. Wang and S. Jegelka. Max-value entropy search for efficient Bayesian optimization. In *International Conference on Machine Learning*, 2017.

[60] Z. Wang, S. Jegelka, L.P. Kaelbling, and T. Lozano-Pérez. Focused model-learning and planning for non-Gaussian continuous state-action systems. In *International Conference on Robotics and Automation*, 2017.

[61] J. Wu and P.I. Frazier. The parallel Knowledge Gradient method for batch Bayesian optimization. In *Advances in Neural Information Processing Systems*, 2016.

A Methods Appendix

A.1 Concrete approximations

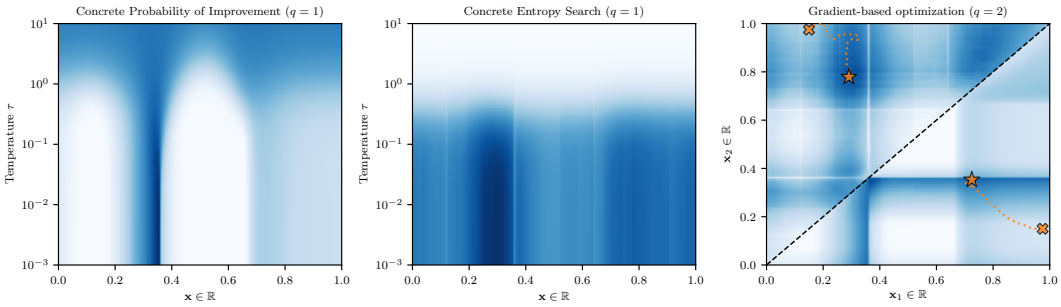

Figure 5: Left: Concrete approximation to PI for temperatures $\tau \in [1\text{e}{-3}, 1\text{e}{-1}]$. Middle: Concrete approximation to ES for temperatures $\tau \in [1\text{e}{-3}, 1\text{e}{-1}]$. Right: Stochastic gradient ascent trajectories when maximizing concrete approximation to parallel versions of PI (top left) and ES (bottom right), both with a temperature $\tau = 0.01$.

As per Section 3.1, utility is sometimes measured in terms of discrete events $e \in \mathcal{E}$. Unfortunately, mappings from continuous values $\mathbf{y} \in \mathcal{Y}$ to the space of discrete events $\mathcal{E}$ are typically discontinuous and, therefore, violate a necessary condition for interchangeability of differentiation and expectation. In the generalized context of differentiating a Monte Carlo integral w.r.t. the parameters of a *discrete* generative distribution, [29, 40] proposed to resolve this issue by introducing a continuous approximation to the aforementioned discontinuous mapping.

As a guiding example, assume that $\boldsymbol{\theta}$ is a self-normalized vector of q parameters such that $\forall \theta \in \boldsymbol{\theta}$, $\theta \geq 0$ and that $\mathbf{z} = [z_1, \ldots, z_q]^\top$ is a corresponding vector of uniform random variables. Subsequently, let element-wise product $\pi(\mathbf{z}; \boldsymbol{\theta}) = \mathbf{z} \circ \boldsymbol{\theta}$ be defined as random variables $\mathbf{y}$'s reparameterization. Denoting by $y^* = \max(\mathbf{y})$, the vector-valued function $\boldsymbol{e} : \mathcal{Y}^q \mapsto \{0, 1\}^q$ defined as

$$\boldsymbol{e}(\mathbf{z}; \boldsymbol{\theta}) = [y_1 \geq y^*, \ldots, y_q \geq y^*]^\top = \mathbb{1}^+ (\mathbf{y} - y^*) \tag{9}$$

then reparameterizes a (one-hot encoded) categorical random variable $\boldsymbol{e}$ having distribution $p(\boldsymbol{e}; \boldsymbol{\theta}) = \prod_{i=1}^q \theta_i^{e_i}$. Importantly, we can rewrite (9) as the zero-temperature limit of as continuous mapping $\tilde{\boldsymbol{e}} : \mathcal{Y}^q \mapsto [0, 1]^q$ defined as

$$\tilde{\boldsymbol{e}}(\mathbf{y}; \tau) = \operatorname{softmax}\left(\frac{\mathbf{y} - y^*}{\tau}\right) = \operatorname{softmax}\left(\frac{\mathbf{y}}{\tau}\right), \tag{10}$$

where $\tau \in [0, \infty]$ is a temperature parameter. For non-zero temperatures $\tau > 0$, we obtain a relaxed version of the original (one-hot encoded) categorical event. Unlike the original however, the relaxed event satisfies the conditions for interchanging differentiation and expectation.

Returning to the case of an acquisition function (1) defined over a multivariate normal belief $p(\mathbf{y}|\mathbf{X})$ with parameters $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$, random variables $\mathbf{y}$ are instead reparameterized by $\pi(\mathbf{z}; \theta) = \boldsymbol{\mu} + \mathbf{L}\, \mathrm{g}(\mathbf{z})$, where g denotes, e.g., the Box-Muller transform of uniform rvs $\mathbf{z} = [z_1, \ldots, z_{2q}]^\top$. This particular example demonstrates how Entropy Search's innermost integrand can be relaxed using a *concrete* approximation.[5] Virtually identical logic can be applied to approximate Probability of Improvement's integrand.

For Monte Carlo versions of both acquisition functions, Figure 5 shows the resulting approximation across a range of temperatures along with gradient-based optimization in the parallel setting $q = 2$. Whereas for high temperatures τ the approximations wash out, both converge to the corresponding true function as $\tau \to 0$.

[5] As in [40], we use "concrete" as a portmanteau of "continuous" and "discrete".

A.2 Parallel Upper Confidence Bound (q-UCB)

For convenience, we begin by reproducing (3) as indefinite integrals,

$$\mathcal{L}(\mathbf{X}) = \int_{-\infty}^{\infty} \ell(\mathbf{y})\mathcal{N}(\mathbf{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma})d\mathbf{y} = \int_{-\infty}^{\infty} \ell(\boldsymbol{\mu} + \mathbf{L}\mathbf{z})\mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})d\mathbf{z}.$$

Working backward through this equation, we derive an exact expression for parallel UCB. To this end, we introduce the definition

$$\sqrt{\frac{\pi}{2}} \int_{-\infty}^{\infty} |\sigma z| \mathcal{N}(z; 0, 1)dz = \sqrt{2\pi} \int_{0}^{\infty} y\mathcal{N}(y; 0, \sigma^2)dy = \sigma, \tag{11}$$

where $|\cdot|$ denotes the (element-wise) absolute value operator.[6] Using this fact and given $z \sim \mathcal{N}(0, 1)$, let $\hat{\sigma}^2 \triangleq (\beta\pi/2)\sigma^2$ such that $\mathbb{E}|\hat{\sigma}z| = \beta^{1/2}\sigma$. Under this notation, marginal UCB can be expressed as

$$\begin{aligned} \text{UCB}(\mathbf{x}; \beta) &= \mu + \beta^{1/2}\sigma \\ &= \int_{-\infty}^{\infty} (\mu + |\hat{\sigma}z|)\, \mathcal{N}(z; 0, 1)dz \\ &= \int_{-\infty}^{\infty} (\mu + |\gamma|)\, \mathcal{N}(\gamma; 0, \hat{\sigma}^2)d\gamma \end{aligned} \tag{12}$$

where (μ, σ^2) parameterize a Gaussian belief over $y = f(\mathbf{x})$ and $\gamma = y - \mu$ denotes y's residual. This integral form of UCB is advantageous precisely because it naturally lends itself to the generalized expression

$$\begin{aligned} \text{UCB}(\mathbf{X}; \beta) &= \int_{-\infty}^{\infty} \max(\boldsymbol{\mu} + |\boldsymbol{\gamma}|)\mathcal{N}(\boldsymbol{\gamma}; \mathbf{0}, \hat{\boldsymbol{\Sigma}})d\boldsymbol{\gamma} \\ &= \int_{-\infty}^{\infty} \max(\boldsymbol{\mu} + |\hat{\mathbf{L}}\mathbf{z}|)\mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})d\mathbf{z} \\ &\approx \frac{1}{n}\sum_{k=1}^{m} \max(\boldsymbol{\mu} + |\hat{\mathbf{L}}\mathbf{z}_k|) \;\text{ for }\; \mathbf{z}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \end{aligned} \tag{13}$$

where $\hat{\mathbf{L}}\hat{\mathbf{L}}^\top = \hat{\boldsymbol{\Sigma}} \triangleq (\beta\pi/2)\boldsymbol{\Sigma}$. This representation has the requisite property that, for any size $q' \leq q$ subset of $\mathbf{X}$, the value obtained when marginalizing out the remaining $q - q'$ terms is its q'-UCB value.

Previous methods for parallelizing UCB have approached the problem by imitating a purely sequential strategy [11, 14]. Because a fully Bayesian approach to sequential selection generally involves an exponential number of posteriors, these works incorporate various well-chosen heuristics for the purpose of efficiently approximate parallel UCB.[7] By directly addressing the associated q-dimensional integral however, Eq. (13) avoids the need for such approximations and, instead, unbiasedly estimates the true value.

Finally, the special case of marginal UCB (12) can be further simplified as

$$\text{UCB}(\mathbf{x}; \beta) = \mu + 2\int_{0}^{\infty} \hat{\sigma}z\mathcal{N}(z; 0, 1)dz = \int_{\mu}^{\infty} y\mathcal{N}(y; \mu, 2\pi\beta\sigma^2)dy, \tag{14}$$

revealing an intuitive form — namely, the expectation of a Gaussian random variable (with rescaled covariance) above its mean.

[6] This definition comes directly from the standard integral identity [23]: $\int_0^b xe^{-q^2x^2}dx = \frac{1-e^{-q^2b^2}}{2q^2}$.

[7] Due to the stochastic nature of the mean updates, the number of posteriors grows exponentially in q.

B Experiments Appendix

B.1 Synthetic tasks

Synthetic tasks: To eliminate model error, experiments were first run on synthetic tasks drawn from a known prior. For a GP with a continuous, stationary kernel, approximate draws f can be constructed via a weighted sum of basis functions sampled from the corresponding Fourier dual [5, 26, 47, 48]. For a Matérn-$\nu/2$ kernel with anisotropic lengthscales $\mathbf{\Lambda}^{-1}$, the associated spectral density is the multivariate t-distribution $t_\nu(\mathbf{0}, \mathbf{\Lambda}^{-1})$ with ν degrees of freedom [36]. For our experiments, we set $\mathbf{\Lambda} = \mathrm{diag}(d/16)$ and approximated f using 2^{14} basis functions, resulting in tasks that were sufficiently challenging and closely resembled exact draws from the prior.

B.2 Multi-start initialization

As noted in [58], gradient-based query optimization strategies are often sensitive to the choice of starting positions. This sensitivity naturally occurs for two primary reasons. First, acquisition functions $\mathcal{L}$ are consistently non-convex. As a result, it is easy for members of query set $\mathbf{X} \subseteq \mathcal{X}$ to get stuck in local regions of the space. Second, acquisition surfaces are frequently patterned by (large) plateaus offering little expected utility. Such plateaus typically emerge when corresponding regions of $\mathcal{X}$ are thought to be inferior and are therefore excluded from the search process.

To combat this issue, we appeal to the submodularity of $\mathcal{L}$ (see Section 3.2). Assuming $\mathcal{L}$ is submodular, then acquisition values exhibit diminishing returns w.r.t. the degree of parallelism q. As a result, the marginal value for querying a single point $\mathbf{x} \in \mathcal{X}$ upper bounds its potential contribution to any query set $\mathbf{X}$ s.t. $\mathbf{x} \in \mathbf{X}$. Moreover, marginal acquisition functions $\mathcal{L}(\mathbf{x})$ are substantially cheaper to compute that parallel ones (see Figure 1d). Accordingly, we can initialize query sets $\mathbf{X}$ by sampling from $\mathcal{L}(\mathbf{x})$. By doing so we gracefully avoid initializing points in excluded regions, mitigating the impact of acquisition plateaus.

In our experiments we observed consistent performance gains when using this strategy in conjunction with most query optimizers. To accommodate runtime constraints, the initialization process was run for the first tenth of the allocated time.

Lastly, when greedily maximizing $\mathcal{L}$ (equiv. in parallel asynchronous cases), "pending" queries were handled by fantasizing observations at their predictive mean. Conditioning on the expected value reduces uncertainty in the vicinity of the corresponding design points and, in turn, promotes diversity within individual query sets [14]. To the extent that this additional step helped in our experiments, the change in performance was rather modest.

B.3 Parallel Upper Confidence Bound (q-UCB)

Additional results obtained using q-UCB are shown here. These experiments were run under identical conditions to those in Section 4. In all cases, we used a confidence parameter $\beta = 2$. Except for on the Levy benchmark, q-UCB outperformed q-EI, and this result also held for both Branin-Hoo and Hartmann-3 (not shown).

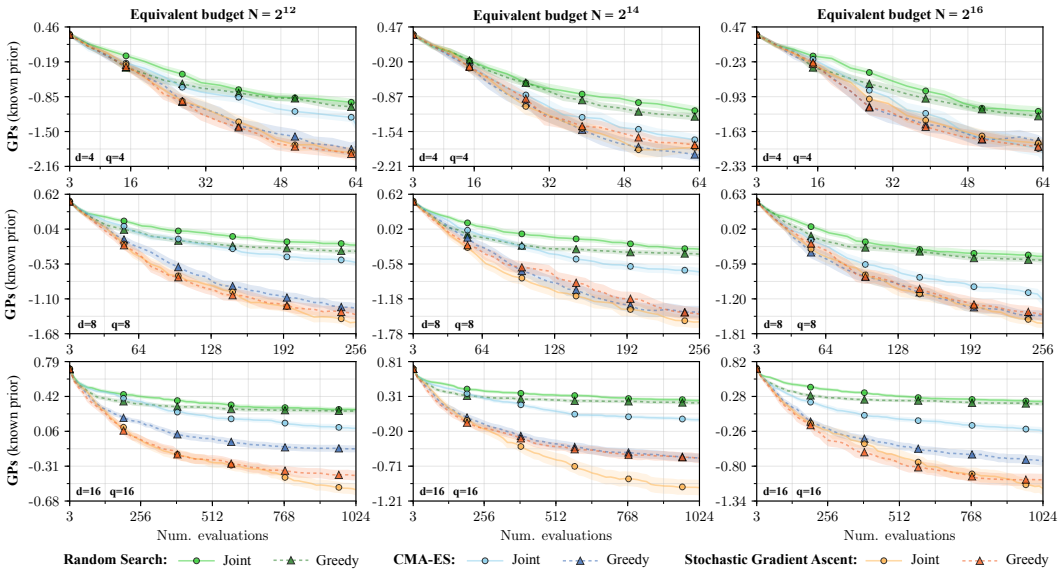

Figure 6: Average performance of different acquisition maximizers on synthetic tasks from a known prior, given varied runtimes when maximizing Monte Carlo q-EI. Reported values indicate the log of the immediate regret $\log_{10} |f_{\max} - f(\mathbf{x}^*)|$, where $\mathbf{x}^*$ denotes the observed maximizer $\mathbf{x}^* \in \arg\max_{\mathbf{x} \in \mathcal{D}} \hat{y}$.

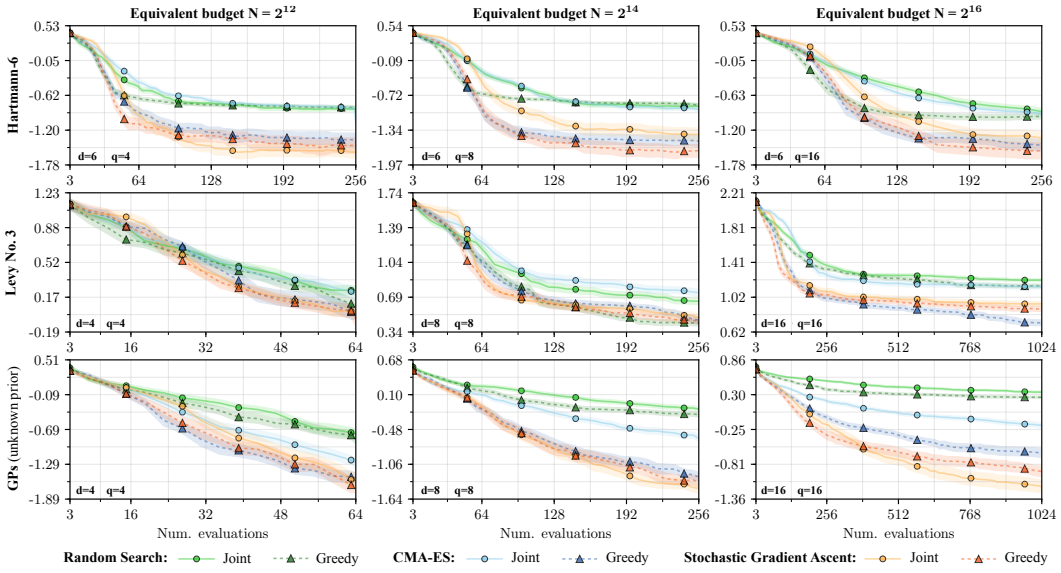

Figure 7: Average performance of different acquisition maximizers on black-box tasks from an unknown prior, given varied runtimes when maximizing Monte Carlo q-EI. Reported values indicate the log of the immediate regret $\log_{10} |f_{\max} - f(\mathbf{x}^*)|$, where $\mathbf{x}^*$ denotes the observed maximizer $\mathbf{x}^* \in \arg\max_{\mathbf{x} \in \mathcal{D}} \hat{y}$.