Gradient-Enhanced Bayesian Optimization via Acquisition Ensembles with Application to Reinforcement Learning [⋆]

Georgios Makrygiorgos [*] **Joel A. Paulson** [**] **Ali Mesbah** [*]

[*] *Department of Chemical and Biomolecular Engineering, University of California, Berkeley, CA 94720, USA (gmakr@berkeley.edu, mesbah@berkeley.edu).*
[**] *Department of Chemical and Biomolecular Engineering, The Ohio State University, Columbus, OH 43210, USA (paulson.82@osu.edu).*

Abstract: Bayesian optimization (BO) has shown great promise as a data-efficient strategy for the global optimization of expensive, black-box functions in a plethora of control applications. Traditional BO is derivative-free, as it solely relies on observations of a performance function to find its optimum. Recently, so-called first-order BO methods have been proposed that additionally exploit gradient information of the performance function to accelerate convergence. First-order BO methods mostly utilize standard acquisition functions, while indirectly using gradient information in the kernel structure to learn more accurate probabilistic surrogates for the performance function. In this work, we present a gradient-enhanced BO method that directly exploits performance function (zeroth-order) and its corresponding gradient (first-order) evaluations in the acquisition function. To this end, a novel gradient-based acquisition function is proposed that can identify fixed points of the performance optimization problem. We then leverage ideas from multi-objective optimization to develop an effective strategy for finding query points that optimally tradeoff between traditional zeorth-order acquisition functions and the proposed gradient-based acquisition function. We show how the proposed acquisition-ensemble, gradient-enhanced BO (AEGEBO) method can be used to accelerate convergence of policy-based reinforcement learning by combining noisy observations of the reward function and its gradient that can be directly estimated from closed-loop data. The performance of AEGBO is compared to traditional BO and the well-known REINFORCE algorithm on a benchmark LQR problem, for which we consistently observe significantly improved performance over a limited data budget.

Keywords: Bayesian optimization; Multi-objective acquisition ensemble; Reinforcement learning

1. INTRODUCTION

In recent years, there has been a growing interest in the use of black-box (or derivative-free) optimization in a variety of real-world control applications. In particular, Bayesian optimization (BO) has emerged as an effective strategy for control-oriented model learning (Bansal et al., 2017; Makrygiorgos et al., 2022), controller auto-tuning (Paulson and Mesbah, 2020; Khosravi et al., 2021; Paulson et al., 2022), and direct policy-search reinforcement learning (Marco et al., 2016, 2017; Pautrat et al., 2018; Turchetta et al., 2020; Chatzilygeroudis et al., 2019). BO is considered especially useful for "global" optimization of black-box and expensive-to-evaluate functions (Frazier, 2018), such as closed-loop control performance measures. BO provides a principled strategy to sequentially query candidate points using an acquisition function (AF), which measures the information value of sampling at a new point in terms of a probabilistic surrogate model of the performance function constructed from previous function observations (i.e., *zeroth-order* information).

Nevertheless, in various optimization and control settings, *first-order* gradient information, namely observations of partial derivatives of performance function with respect to decision variables, is readily available. Enhancing the convergence rate of BO using gradient information has been investigated in a few recent works (Wu et al., 2017; Eriksson et al., 2018; Shekhar and Javidi, 2021). The main idea is to condition the GP model of performance function on gradient information to obtain more accurate predictions, which can in turn yield faster convergence. As such, the first-order gradient information is used indirectly in searching for the candidate sample points. More recently, the direct use of gradients in the search process has been proposed mainly in the context of policy-search reinforcement learning (RL) to locally enhance the performance of gradient descent. To this end, Müller et al. (2021) and Nguyen et al. (2022) have utilized AFs based on only first-order information to obtain improved gradient estimates for a black-box reward function using gradients of a GP model. Penubothula et al. (2021) proposed a first-order BO method that uses a collection of AFs built separately for each partial derivative.

[⋆] G. Makrygiorgos and J.A. Paulson have equally contributed to this work. The work was supported by the US National Science Foundation under grant 2130734.

A clustering method is then used to find a consensus point via a convex combination of the set optimal points found for each individual AF. Not only does this method require several acquisition functions to be maximized at each iteration, the heuristic clustering method is not guaranteed to optimally tradeoff between these different AFs. In the context of traditional zeroth-order BO, recent work has shown the value of applying multi-objective optimization techniques to systematically identify Pareto optimal points for an ensemble of AFs (Lyu et al., 2018; Guinet et al., 2020). To the best of our knowledge, such concepts have yet to be applied in any first-order BO method.

In this work, we present a gradient-enhanced BO method that can exploit both performance and gradient function evaluations using an ensemble of two acquisition functions. The contribution of this paper is twofold. The first is the derivation of a cheap-to-evaluate gradient-based acquisition function that can identify fixed points of the performance optimization problem. The second contribution is a simple, yet effective strategy for finding query points that optimally tradeoff between a zeroth-order AF and the proposed gradient-based AF via multi-objective optimization. Thus, the proposed acquisition-ensemble, gradient-enhanced BO (AEGBO) method can discover a set of valuable query points that are Pareto optimal with respect to both sources of information. Furthermore, we discuss how AEGBO can be applied to policy-search RL to accelerate convergence by using noisy observations of reward function and its gradient, which can be directly estimated from closed-loop observations of the reward function using the policy gradient theorem (Sutton et al., 1999). The performance of AEGBO is compared to standard BO and the well-known REINFORCE algorithm on a benchmark LQR problem.

2. NOTATION AND PRELIMINARIES

2.1 Problem Statement

Given an expensive-to-evaluate function $f : \mathbb{X} \to \mathbb{R}$, we are interested in finding the design (or input) vector $x^\star$ that globally maximizes the function, i.e.,

$$x^\star \in \operatorname*{argmax}_{x \in \mathbb{X}} f(x), \tag{1}$$

where $\mathbb{X} \subset \mathbb{R}^d$ is the optimization domain. The mathematical structure of f is assumed to be unknown such that we must rely on some "learning" strategy to infer a representation of the function from data. To execute the learning process, we assume that we have the ability to query f at any desired input $x \in \mathbb{X}$ and receive a (possibly noisy) evaluation of $f(x)$ and its gradient $\nabla f(x)$. Traditional BO methods consider zeroth-order (derivative-free) function evaluations only, which fundamentally limits performance when additional gradient information is available. The goal of this work is to simultaneously utilize zeroth- and first-order information in a computationally efficient manner. We first provide a short overview of previous BO methods that use function and gradient information.

2.2 Gaussian Processes with Derivative Information

We place a GP prior over f to build a probabilistic surrogate model that is non-parametric. A GP model is fully specified by its mean function $\mu : \mathbb{X} \to \mathbb{R}$ and covariance (or kernel) function $k : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$. Since the gradient is a linear operator, the gradient of a GP must remain a GP, such that we can create a joint GP model with the following updated mean function $\tilde{\mu}$ and covariance function $\tilde{k}$

$$\tilde{\mu}(x) = \begin{bmatrix} \mu(x) \\ \nabla \mu(x) \end{bmatrix}, \tag{2a}$$

$$\tilde{k}(x, x') = \begin{bmatrix} k(x, x') & \nabla_{x'} k(x, x')^\top \\ \nabla_x k(x, x') & \nabla_x (\nabla_{x'} k(x, x')^\top) \end{bmatrix}. \tag{2b}$$

The extended mean function $\tilde{\mu} : \mathbb{X} \to \mathbb{R}^{d+1}$ maps to a $(d+1)$-dimensional vector, while the extended covariance function $\tilde{k} : \mathbb{X} \times \mathbb{X} \to \mathbb{R}^{d+1 \times d+1}$ maps to a $(d+1) \times (d+1)$ matrix, which has the ability to capture correlation between the function and its d partial derivatives that make up the gradient vector (Williams and Rasmussen, 2006, Sect. 9.4). We assume access to some dataset $\mathcal{D}^{(n)} = \{(x^{(i)}, y^{(i)}, \nabla y^{(i)})\}_{i=1}^n$ composed of n sample points with corresponding noisy observations of the objective and gradient at each $x^{(i)}$ given by

$$(y^{(i)}, \nabla y^{(i)}) \sim \mathcal{N}\left((f(x^{(i)}), \nabla f(x^{(i)})), \Sigma^{(i)} \right), \tag{3}$$

where $\Sigma^{(i)} \in \mathbb{R}^{d+1 \times d+1}$ is a positive-definite covariance matrix for the i^{th} sample point. If $\Sigma^{(i)}$ is not known, then we typically parametrize it as $\Sigma^{(i)} = \text{diag}\left(\sigma_1^2, \ldots, \sigma_{d+1}^2\right)$, where σ_k^2 denotes a fixed independent variance term for each separate element of the observation vector $k \in \{1, \ldots, d+1\}$, which can be estimated from data.

Given the current dataset $\mathcal{D}^{(n)}$, the posterior $(f, \nabla f) \mid \mathcal{D}^{(n)}$ remains a joint GP with the following updated mean function $\tilde{\mu}^{(n)}$ and covariance function $\tilde{k}^{(n)}$

$$\tilde{\mu}^{(n)}(x) = \tilde{\mu}(x) + \tilde{\boldsymbol{k}}_n^\top(x) \tilde{\boldsymbol{K}}_n^{-1} (\tilde{\boldsymbol{y}}_n - \tilde{\boldsymbol{\mu}}_n), \tag{4a}$$

$$\tilde{k}^{(n)}(x, x') = \tilde{k}(x, x') - \tilde{\boldsymbol{k}}_n^\top(x) \tilde{\boldsymbol{K}}_n^{-1} \tilde{\boldsymbol{k}}_n(x), \tag{4b}$$

where $\tilde{\boldsymbol{k}}_n(x) = [\tilde{k}(x^{(1)}, x), \ldots, \tilde{k}(x^{(n)}, x)]^\top$ is the vector of covariance values between the sample points and the test point x, $\tilde{\boldsymbol{K}}_n$ is the covariance matrix evaluated at the sample points, which is composed of elements $[\tilde{\boldsymbol{K}}_n]_{ij} = \tilde{k}(x^{(i)}, x^{(j)}) + \Sigma^{(i)} \delta_{ij}$, $\tilde{\boldsymbol{y}}_n = ((y^{(1)}, \nabla y^{(1)}), \ldots, (y^{(n)}, \nabla y^{(n)}))$ is a concatenated vector of all observations, and $\tilde{\boldsymbol{\mu}}_n = (\tilde{\mu}(x^{(1)}), \ldots, \tilde{\mu}(x^{(n)}))$ is the joint mean function evaluated at the sample points. Note that the derivative-informed GP approach can straightforwardly extended to the case of incomplete observations by removing the corresponding rows and columns in (4).

2.3 Derivative-enabled Acquisition Functions

Given the probabilistic surrogate model in (4), we must define an acquisition function $\alpha^{(n)} : \mathbb{X} \to \mathbb{R}$ to provide a good measure of the (expected) desirability of querying any point $x \in \mathbb{X}$ with respect to our end goal (maximizing the unknown function f). If properly selected, one would like to preferentially sample at the point that produces the highest value of the acquisition function. We can then formally define BO as the sequential learning process of selecting next samples in the following fashion

$$x^{(n+1)} \in \operatorname*{argmax}_{x \in \mathbb{X}} \alpha^{(n)}(x), \tag{5}$$

where $\alpha^{(n)}(\cdot)$ represents the acquisition function induced by the posterior conditioned on data $\mathcal{D}^{(n)}$. Therefore, the main distinction between traditional BO and gradient-enhanced BO is that $\mathcal{D}^{(n)}$ includes derivative information for the latter, which necessitates the use of a more complex GP model. In principle, one could take advantage of any of the previously developed acquisition functions (Shahriari et al., 2015), such as expected improvement (EI), upper confidence bound (UCB), or knowledge gradient (KG), by replacing the standard posterior mean and variance predictions for f with those derived from (4). However, performing hyperparameter training and posterior update using (4) can be computationally demanding. In particular, inverting the covariance matrix $\tilde{\boldsymbol{K}}_n$ for the joint GP model scales as $O((n(d+1))^3)$, which can be challenging when either n or d are modest in size. Furthermore, this additional cost can have a big impact on the effort needed to solve (5), which requires repeated forward predictions to be made with the joint GP model.

To better understand the computational implications, let us look at the derivative-enabled KG (dKG) function in more detail, which is one of the most commonly used acquisitions in the noisy problem setting considered in this work. The dKG function is defined as follows (Wu et al., 2017)

$$\text{dKG}_n(x) = \mathbb{E}_n\left[\max_{x\in\mathbb{X}} \tilde{\mu}_1^{(n+1)}(x) | x^{(n+1)} = x\right] - \max_{x\in\mathbb{X}} \tilde{\mu}_1^{(n)}(x)$$

where $\tilde{\mu}_1^{(n)}(x) = \mathbb{E}_n\{f(x)\}$ is the expectation of $f(x)$ under the GP prior conditioned on data $\mathcal{D}^{(n)}$ (corresponds to the first element of the joint mean vector $\tilde{\mu}^{(n)}(x)$). Thus, dKG measures the expected improvement in the maximum value of the mean function given a new observation is taken at $x_{n+1} = x$. The use of the mean function, as opposed to the function observations themselves, is needed to filter out any noise present in the observations. Although dKG is a quite effective measure of the value of information, it is very expensive to evaluate due to the internal maximization over the future posterior mean function. To mitigate this computational burden, (Wu et al., 2017) proposed to only use the best directional derivative at each iteration. In addition to ignoring useful information in the form of the complete set of partial derivatives of the objective function, this approach does not fully address the inherent challenge of the two-level optimization procedure needed to globally solve (5) when $\alpha^{(n)}(x) = \text{dKG}_n(x)$.

In the next section, we introduce a new strategy that can overcome the challenges of previously developed derivative-enhanced acquisition functions (such as dKG). Not only does the proposed method take advantage of the complete set of function and gradient information at each iteration, it requires significantly less computational cost than currently available methods.

3. ACQUISITION ENSEMBLE WITH GRADIENTS BAYESIAN OPTIMIZATION (AEGBO)

In this section, we describe the proposed method for efficiently integrating noisy function and gradient information into the BO framework, referred to as AEGBO. First, we discuss a simpler model for the function and its gradient constructed from a set of independent GPs. Then, we introduce a novel acquisition function that can identify points likely to satisfy $\nabla f(x) = 0$, which is a necessary condition for optimality of (1). Lastly, we present a strategy to select a point for which both a zeroth-order (function) and first-order (gradient) acquisition function agree is valuable using ideas from multi-objective optimization.

3.1 Independent Gaussian Process Models

Instead of using the complete joint GP model (4), we treat the objective function and each one of its partial derivatives as independent. This implies that the GP priors are completely separate, i.e.,

$$f(x) \sim \mathcal{GP}(\mu_0(x), k_0(x, x')), \tag{6a}$$

$$\frac{\partial f(x)}{\partial x_i} \sim \mathcal{GP}(\mu_i(x), k_i(x, x')), \quad \forall i \in \{1, \dots, d\}, \tag{6b}$$

where μ_0 and k_0 correspond to the mean and kernel functions for the function itself, respectively, and μ_i and k_i correspond to the mean and kernel functions for the i^{th} partial derivative of the function, respectively. Let $\mathcal{D}^{(n)} = \{\mathcal{D}_0^{(n)}, \mathcal{D}_1^{(n)}, \dots, \mathcal{D}_d^{(n)}\}$ be divided into datasets corresponding to the function observations $\mathcal{D}_0^{(n)}$ and each of the partial derivative observations $\{\mathcal{D}_i^{(n)}\}_{i=1}^d$. Then, due to the independence assumption, we can construct the posterior mean and kernel functions for each of the $(d+1)$ GP models, denoted by $\mu_i^{(n)}(x)$ and $k_i^{(n)}(x, x')$, separately using only the local data $\mathcal{D}_i^{(n)}$ for all $i = 0, \dots, d$. The posterior update equations are analogous to (4), except the operations are only performed on a subset of data, implying the computational cost has been reduced to $O((d+1)n^3)$, which is linear with respect to d. Furthermore, these operations can be carried out in parallel, which would make the cost independent of d. Another advantage of the independent GPs in (6) is that the estimated noise values for each function are also independent – we have found this to be very important in practice to reduce sensitivity of the overall predictions to the estimated noise levels.

3.2 Novel Gradient-based Acquisition Function

Here, we focus on UCB-style acquisition functions due to their simplicity and established convergence properties (Lu and Paulson, 2022). The UCB function is given by

$$\alpha_{\text{UCB}}^{(n)}(x) = \mu_0^{(n)}(x) + \beta_f \sigma_0^{(n)}(x), \tag{7}$$

where $\beta_f \in \mathbb{R}_+$ is a hyperparameter that balances exploration (information) and exploitation (performance) and $\sigma_0^{(n)}(x) = [k_0^{(n)}(x, x)]^{1/2}$ is the standard deviation of the posterior GP for f.

Under the independence assumption, the gradient predictions do not directly impact the UCB acquisition such that we need a new strategy for quantifying the value of gradient information. To derive an independent source of information, we recognize that a necessary condition for optimality in (1) is $\nabla f(x) = 0$. An equivalent way to represent the solutions to this set of equations is $\min_{x\in\mathbb{X}} \|\nabla f(x)\|$, which can also be stated as $\max_{x\in\mathbb{X}}(-\|\nabla f(x)\|)$, where $\|\cdot\|$ denotes some vector norm (we focus on the 1-norm here for simplicity). Since the gradient is also an unknown function, we can try to use BO methods to tackle this optimization problem as a way to efficiently search for fixed points of the original maximization problem of interest. An important distinction between the gradient norm (GN) problem and (1) is that the former involves multiple unknown functions. This is often referred to as a decomposed BO problem in the literature for which standard acquisition functions do not directly apply. We can straightforwardly develop an UCB acquisition function for multi-output problems whenever the objective is defined as a linear transformation of the GP models, as shown in (Kudva et al., 2022).

Since norms are nonlinear operators, however, we need a tailored approximation strategy for the gradient norm case of interest here. We propose the following gradient-based acquisition function analogously to the UCB function above

$$\alpha_{\text{GN}}^{(n)}(x) = -\mathbb{E}_n\{\|\nabla f(x)\|\} + \beta_g \sqrt{\text{Var}_n\{\|\nabla f(x)\|\}}, \quad (8)$$

where $\text{Var}_n\{\cdot\}$ denotes the posterior variance given $\mathcal{D}_n$ and $\beta_g \in \mathbb{R}_+$ is a hyperparmeter similar to β_f. It turns out we can construct analytic expressions for the mean and variance terms using properties of Gaussian random variables. Starting with the mean term, we can derive

$$\begin{aligned}\mathbb{E}_n\{\|\nabla f(x)\|\} &= \textstyle\sum_{i=1}^{d} \mathbb{E}_n \left\{ \left| \frac{\partial f(x)}{\partial x_i} \right| \right\}, \quad (9)\\ &= \sum_{i=1}^{d} \left\{ 2\sigma_i^{(n)}(x)\phi(z_i^{(n)}) + \mu_i^{(n)}(x) \left[\Phi(z_i^{(n)}) + \Phi(z_i^{(n)}) \right] \right\},\end{aligned}$$

where $z_i^{(n)} = \mu_i^{(n)}(x)/\sigma_i^{(n)}(x)$ and $\phi(\cdot)$ and $\Phi(\cdot)$ correspond to the standard normal density function and cumulative density function, respectively. We can similarly derive a simple overall expression for the variance

$$\begin{aligned}\text{Var}_n\{\|\nabla f(x)\| &= \textstyle\sum_{i=1}^{d} \text{Var}_n \left\{ \left| \frac{\partial f(x)}{\partial x_i} \right| \right\}, \quad (10)\\ &= \sum_{i=1}^{d} \left\{ (\mu_i^{(n)}(x))^2 + (\sigma_i^{(n)}(x))^2 - \mathbb{E}_n \left\{ \left| \frac{\partial f(x)}{\partial x_i} \right| \right\}^2 \right\},\end{aligned}$$

where we have exploited the independence assumption in the first line and rewrote the individual variance terms in terms of the second total moment in the second line. Note that closed-form expressions for the inner expectation terms have already been computed in (9). As such, our proposed gradient-based acquisition function in (8) can be efficiently computed using the d separate GP models for each of the partial derivatives of the objective function – this implies that maximizing $\alpha_{\text{GN}}^{(n)}(x)$ should be at worst a linear factor of the cost required to maximize the cheap-to-evaluate function $\alpha_{\text{UCB}}^{(n)}(x)$ with respect to d. This is a substantial reduction in cost when compared to the dKG function described previously.

3.3 Combining Function and Gradient Information using Acquisition Ensembles

Now we are equipped with two separate acquisition functions $\alpha_{\text{UCB}}^{(n)}(x)$ and $\alpha_{\text{GN}}^{(n)}(x)$ that, respectively, provide independent sources of zeroth- and first-order information regarding the maxima of f. It is unlikely that the same point maximizes both of these functions simultaneously, meaning we need some procedure to select a common value x_{n+1} that performs reasonably well with respect to both functions. The multi-objective optimization (MOO) framework is a good fit for this task since it allows us to systematically tradeoff between multiple objectives. The main goal of MOO is to characterize the set of points on the so-called *Pareto frontier*, which is the set of Pareto optimal points, i.e., feasible points $x \in \mathbb{X}$ in which favorable movement in one objective comes at the expense of at least one other objective. In (Chen et al., 2022), a related idea is applied to a set of traditional BO acquisition functions that showed promising results. Therefore, we look to develop a similar approach using $\boldsymbol{\alpha}_n(x) = \{\alpha_1^{(n)}(x), \alpha_2^{(n)}(x)\}$ as our set of acquisition functions, where the subscripts 1 and 2 will be used as shorthand for the UCB and GN acquisition functions, respectively. Therefore, we can now formally state the AEGBO method in terms of $\boldsymbol{\alpha}_n(x)$ as the following sequential sampling process

$$x^{(n+1)} \in X_n^\star = \{x \in \mathbb{X} : \boldsymbol{\alpha}_n(x) \in \mathcal{P}_n\}, \quad (11)$$

where $X_n^\star$ denotes the set of Pareto optimal points given all currently available data $\mathcal{D}^{(n)}$, which is characterized by the Pareto frontier $\mathcal{P}_n$

$$\mathcal{P}_n = \{\boldsymbol{\alpha}_n(x) : \nexists y \in \mathbb{X} \text{ s.t. } \boldsymbol{\alpha}_n(x) \prec \boldsymbol{\alpha}_n(y)\}, \quad (12)$$

where $\boldsymbol{\alpha}_n(x) \prec \boldsymbol{\alpha}_n(y)$ implies point y *dominates* x, which occurs if and only if $\alpha_i^{(n)}(x) \leq \alpha_i^{(n)}(y)$ for all $i \in \{1,2\}$ and $\exists i \in 1,2$ such that $\alpha_i^{(n)}(x) < \alpha_i^{(n)}(y)$. Therefore, $\mathcal{P}_n$ corresponds to the set of points for which there does not exist any feasible point that dominates it.

Although the proposed AEGBO method requires the MOO problem (11) be solved at every iteration, this problem involves only two cheap-to-evaluate objective functions and thus can be straightforwardly solved (approximately) using established methods such as the NSGA-II algorithm (Deb et al., 2002). It is worth noting that all points in $X_n^\star$ are Pareto optimal such that there is no clear metric to select between the candidate points in this set. In practice, any selection criteria can be utilized; however, we focus on a uniform random selection criteria in which all points from $X_n^\star$ are potentially chosen with equal probability. We found that such an approach tends to reduce bias that may be introduced with a deterministic selection strategy. Additionally, since the hypervolume of $\mathcal{P}_n$ tends to decrease as more data is collected, the randomized approach is unlikely to over-explore as $n \to \infty$.

4. AEGBO FOR POLICY-BASED REINFORCEMENT LEARNING OF EXPENSIVE SYSTEMS

Reinforcement learning (RL) is a semi-supervised learning method in which a so-called "agent" attempts to learn the best way to maximize a long-term reward function through trial-and-error interactions with the "environment." There has been a vast amount of work on RL, which can be roughly viewed as a collection of solution approaches to stochastic optimal control problems of the form

$$\begin{aligned}\max_{\pi_{0:N-1}} \quad & \mathbb{E}_{w_{0:N-1}} \left\{ \textstyle\sum_{t=0}^{N-1} r_t(z_t, u_t, w_t) + r_N(z_N) \right\}, \quad (13)\\ \text{s.t.} \quad & z_{t+1} = g_t(z_t, u_t, w_t), \quad u_t = \pi_t(\tau_t),\end{aligned}$$

when the system dynamics are unknown, where z_t, u_t, and w_t are the system state, control input, and disturbance at time t, respectively, $g_t(\cdot)$ is the state transition function that governs the dynamics at time t in response to stochastic disturbances w_t, $r_t(\cdot)$ is the reward gained at time step t, $\pi_t(\cdot)$ is the feedback control policy at time t that can be any feasible function of the observed data trajectory up until time t, i.e., $\tau_t = (u_0, \dots, u_{t-1}, x_0, \dots, x_t)$, and N is the time horizon. Thus, RL methods attempt to solve (13) in cases where the state transition rules $\{g_t(\cdot)\}$ are unknown by transforming the problem into a learning task. We can then classify different RL methods based on the chosen learning task (see, e.g., (Mesbah et al., 2022)).

One of the most popular variants in recent years is the so-called class of policy-based RL methods that look to learn the optimal settings for a parametrized policy function $\pi_x(\tau_t)$ where x refers to adjustable policy parameters. We can think of the overall reward function in (13) as simply a function of x since this represents the only degrees of freedom remaining in the policy. Let us define $R(\tau)$ as the overall reward function computed over a single dynamic trajectory τ. Due to the random disturbances present in the dynamics, τ is random with some probability distribution $p(\tau; x)$ that is parametrized by x such that (13) becomes

$$\max_{x \in \mathbb{X}} \; f(x) = \mathbb{E}_{p(\tau;x)}\{R(\tau)\} = \int R(\tau)p(\tau; x)d\tau, \qquad (14)$$

which matches our starting problem (1) since f is unknown. The first key observation is that noisy observations are critically important to handle in policy-based RL problems since we cannot evaluate the integral in (14) exactly and must resort to some sampling strategy, e.g., $\frac{1}{N_s}\sum_{i=1}^{N_s} R(\tau^{(i)})$ where $\tau^{(i)} \sim p(\tau; x)$. Traditional BO methods can be applied in such cases, however, this only takes advantage of zeroth-order information. Policy gradient methods are a commonly used alternative that exploit the fact that gradient estimates of the reward can be derived as follows

$$\nabla f(x) = \mathbb{E}_{p(\tau;x)}\{R(\tau)\nabla_x \log p(\tau; x)\}, \qquad (15)$$

which can be evaluated using only gradients of the policy for Markov processes (Sutton et al., 1999). Traditional policy gradient methods, such as REINFORCE (Williams, 1992), then apply stochastic gradient ascent to update an initial $x^{(0)}$ using a mini-batch of samples, i.e.,

$$x^{(n+1)} = x^{(n)} + \frac{\eta_n}{N_s}\left(\sum_{i=1}^{N_s} R(\tau^{(i)})\nabla_x \log p(\tau^{(i)}; x)\right), \qquad (16)$$

where η_n is the step size at iteration n (sometimes referred to as a learning rate). However, as observed from (16), these types of policy gradient methods only use estimates of the current gradient at each iteration, which neglects valuable information about the current and past reward and gradient estimates. For example, if the current reward value is small, we should prioritize moving to a new region of the policy parameter space, as opposed to wasting closed-loop evaluations at neighboring parameter values that are likely to perform poorly. Such an efficient sampling strategy is extremely important whenever the closed-loop data collection process is expensive, which will be the case when the system dynamics are defined in terms of a high-fidelity simulator or a time-consuming experimental setup.

The proposed AEGBO method in Section 3 is perfectly suited to take advantage of the complete history of reward and its gradient evaluations at every iteration. Therefore, we can think of AEGBO as a powerful hybrid strategy that inherits the efficient global search capability of BO as well as the useful local search behavior of REINFORCE. It is worth noting that the cost of evaluating $\nabla_x \log p(\tau^{(i)}; x)$ is roughly the same as a single evaluation of the policy itself, meaning the gradient estimate can be obtained for free whenever the reward evaluation $R(\tau^{(i)})$ is much more expensive than the policy evaluation $\pi_x(\tau^{(i)})$, which will often be the case for expensive systems.

Remark 1. The gradient estimate used in (16) is noisy in practice, which has led to a substantial amount of work on ways to reduce the variance of this estimator. We apply the method from (Papini et al., 2018) in our case study, which involves subtracting a baseline value derived from the generated sample trajectories, as we found this approach worked well in our case study. Since the noise levels can have a big influence on the performance of any BO method, we plan to study this in more detail in future work.

5. ILLUSTRATIVE EXAMPLE: LINEAR QUADRATIC REGULATOR

5.1 System Description

To demonstrate the achievable performance gains with AEGBO, we consider a linear quadratic regulator (LQR) problem of the form (13) with a quadratic reward function $r_t(z_t, u_t, w_t) = -z_t^\top Q z_t - u_t^\top R u_t$, a linear system dynamic $z_{t+1} = Az_t + Bu_t + w_t$ with $w_t \sim \mathcal{N}(0, 10^{-2}I)$, no terminal cost, and a time horizon of $N = 10$. The true values for (A, B, Q, R) are given by

$$A = 0.5\begin{bmatrix} 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 2 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \; B = \begin{bmatrix} 0.5 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \; Q = 10^{-2}I, \; R = 10^{-2}.$$

The initial condition is $z_0 = [2.0, -1.5, -2.0, 1.0]^\top$. When the system dynamics are known, the LQR problem can be analytically solved using dynamic programming. For the settings considered here, the optimal control policy as $N \to \infty$ is $\pi^\star(z) = -Kz$ where $K = [1.172, 0.011, 1.516, 1.469]$. This corresponds to an optimal reward value of -0.246.

5.2 Implementation Details

In the context of RL, the system dynamics are assumed unknown such that we need to repeatedly interact with the system to learn a suitable control policy. As discussed in Section 4, we focus on policy-based RL methods and assume a stochastic linear policy function of the form

$$\pi_x(z_t) = \mathcal{N}(-x^\top z_t, \sigma^2), \qquad (17)$$

where $x \in \mathbb{X} = [0, 2]^4 \subset \mathbb{R}^4$ are the policy parameters and $\sigma^2 = 10^{-2}$ is a small variance term needed to ensure the policy gradient theorem used to derive (15) holds.

To implement AEGBO, we use the `fitgpr` and `gamultiobj` functions in Matlab to, respectively, train the hyperparameters of the GP models (assuming a standard zero prior mean function and squared exponential kernel) and identify the set of Pareto optimal points $X_n^\star$ needed in (11). We use a "mini-batch" size of $N_s = 16$ samples during each episode (training epoch) to estimate the reward and gradient values. We selected exploration parameters of $\beta_f = 0.1$ and $\beta_g = 0$.

5.3 Results and Performance Comparisons

We compare AEGBO to two baseline algorithms on the LQR problem to demonstrate its performance improvements. Since our goal is to identify the policy parameters that maximize the reward function in as few iterations as possible, we use simple regret as our performance metric

$$\text{Regret}_n(\mathcal{D}^{(0)}) = f^\star - \max_{i=1,\ldots,n} y^{(i)}, \tag{18}$$

where $f^\star = \max_{x\in\mathbb{X}} f(x)$ is the true global maximum. By definition, simple regret measures the distance between the best observed point and the true solution, which depends on the initial dataset $\mathcal{D}^{(0)}$. Here, we assume that $\mathcal{D}^{(0)}$ is composed of 5 points chosen uniformly at random from the design space $\mathbb{X}$. We estimate average performance $\mathbb{E}\{\text{Regret}_n(\mathcal{D}^{(0)})\}$ by repeating the algorithms 100 times for different $\mathcal{D}^{(0)}$ and report corresponding confidence intervals using the standard error formula (i.e., standard deviation divided by the square root of the number of samples). The two baseline algorithms are described next.

BO: The sampled point is $x^{(n+1)} \in \text{argmax}_{x\in\mathbb{X}} \alpha^{(n)}_{\text{UCB}}(x)$, which only considers zeroth-order information. We keep all other settings the same as that used in AEGBO.

REINFORCE: The REINFORCE algorithm corresponds to the stochastic gradient ascent update step shown in (16), which uses only local first-order information at every iteration. We set the learning rate $\eta_n = 0.1$, which is a commonly used default value (and is the same order as the exploration parameters used in BO and AEGBO).

The average simple regret versus the number of iterations (or episodes e for short) is shown in Fig. 1. We see that AEGBO consistently outperforms BO and REINFORCE within 30 total closed-loop episodes, achieving more than one order of magnitude reduction in simple regret by iteration 30. Furthermore, AEGBO shows a steady reduction in simple regret after every episode, implying it can more consistently identify policy parameters that increase the reward. REINFORCE, on the other hand, shows an initial fast drop in regret, but its convergence rate quickly slows down. It is also worth noting that REINFORCE would be expected to show much worse performance on more challenging problems that contain multiple local optima since it is prone to getting stuck in local solutions. Therefore, this result highlights the value of combining local and global information in the decision-making process.

To better understand the underlying source of AEGBO's improved performance, Fig. 2 shows the evolution of the Pareto frontier in (12) at different episode values. In the early episodes, we see that Pareto frontier is fairly elongated since there is a significant amount of uncertainty in the GP predictions. This implies there is significant mismatch between the points that may lead to large reward values and those that are likely to satisfy the necessary optimality conditions given our current information. As more data is collected, we see that the Pareto frontier begins to shrink substantially, indicating lower uncertainty in the predicted maximum point. Furthermore, we see that the proposed GN acquisition function provides us with an independent source of information that helps select high reward points that are also likely to satisfy $\nabla f(x) = 0$. Looking at $e = 20$, for example, we see that several points are predicted to perfectly satisfy the necessary optimality condition while simultaneously having large reward values. Thus, the fusion of zeroth- and first-order information appears to be at the heart of the improved performance observed in Fig. 1.

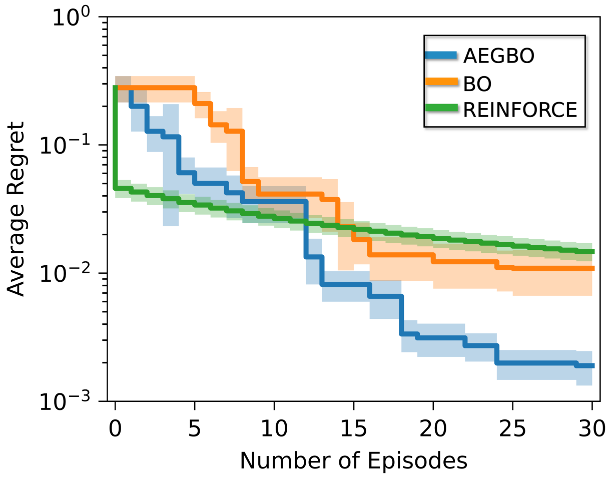

Fig. 1. Expected simple regret (solid lines) and corresponding confidence bounds (shaded regions), estimated using 100 independent realizations of the initial dataset, over 30 closed-loop episodes for AEGBO, traditional BO, and REINFORCE.

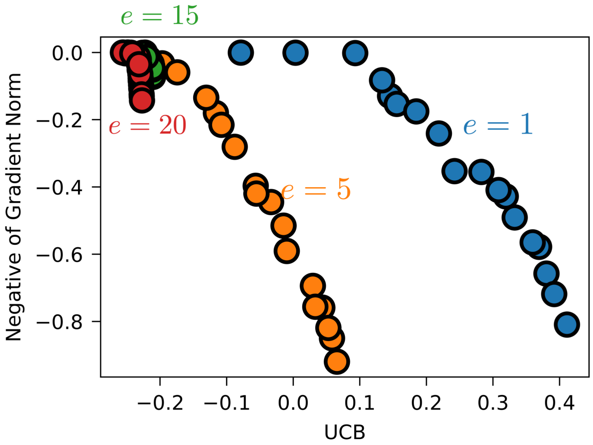

Fig. 2. Pareto frontiers for the proposed MOO acquisition ensemble method at four different closed-loop episodes $e \in \{1, 5, 15, 20\}$ for a representative AEGBO run. The x-axis corresponds to the zeroth-order UCB acquisition function in (7) and the y-axis corresponds to the first-order GN acquisition function in (8).

6. CONCLUSIONS AND FUTURE WORK

This paper presents a gradient-enhanced Bayesian optimization (BO) method, referred to as AEGBO, that can simultaneously exploit evaluations of performance function and its gradients. AEGBO is composed of two key parts: (i) a new first-order acquisition function that quantifies the likelihood of a future query point satisfying necessary optimality conditions and (ii) a multi-objective optimization approach for combining zeroth- and first-order information to accelerate convergence toward the global solution. We also discuss how AEGBO can be applied to policy-search reinforcement learning (RL) problems at virtually no additional cost over traditional BO. The proposed AEGBO method is demonstrated on a RL problem inspired from LQR, where the goal is to identify optimal policy parameters using as little closed-loop data as possible. We show that AEGBO quickly identifies near-optimal solutions in significantly less iterations than state-of-the-art alternative methods. There are several interesting directions for future work including better understanding of the theoretical properties of AEGBO, incorporation of black-box constraints, and its application to more complex RL problems.

REFERENCES

Bansal, S., Calandra, R., Xiao, T., Levine, S., and Tomlin, C.J. (2017). Goal-driven dynamics learning via Bayesian optimization. In *Proceedings of the 56th IEEE Conference on Decision and Control*, 5168–5173. Miami.

Chatzilygeroudis, K., Vassiliades, V., Stulp, F., Calinon, S., and Mouret, J.B. (2019). A survey on policy search algorithms for learning robot controllers in a handful of trials. *IEEE Transactions on Robotics*, 36(2), 328–347.

Chen, J., Luo, F., and Wang, Z. (2022). Dynamic multi-objective ensemble of acquisition functions in batch Bayesian optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 479–482. Boston.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197.

Eriksson, D., Dong, K., Lee, E., Bindel, D., and Wilson, A.G. (2018). Scaling gaussian process regression with derivatives. *Advances in neural information processing systems*, 31.

Frazier, P.I. (2018). A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*.

Guinet, G., Perrone, V., and Archambeau, C. (2020). Pareto-efficient acquisition functions for cost-aware Bayesian optimization. *arXiv preprint arXiv:2011.11456*.

Khosravi, M., Behrunani, V.N., Myszkorowski, P., Smith, R.S., Rupenyan, A., and Lygeros, J. (2021). Performance-driven cascade controller tuning with Bayesian optimization. *IEEE Transactions on Industrial Electronics*, 69(1), 1032–1042.

Kudva, A., Sorouifar, F., and Paulson, J.A. (2022). Efficient robust global optimization for simulation-based problems using decomposed Gaussian processes: Application to MPC calibration. In *Proceedings of the American Control Conference*, 2091–2097. Atlanta.

Lu, C. and Paulson, J.A. (2022). No-regret Bayesian optimization with unknown equality and inequality constraints using exact penalty functions. *IFAC-PapersOnLine*, 55(7), 895–902.

Lyu, W., Yang, F., Yan, C., Zhou, D., and Zeng, X. (2018). Batch Bayesian optimization via multi-objective acquisition ensemble for automated analog circuit design. In *International conference on machine learning*, 3306–3314.

Makrygiorgos, G., Bonzanini, A.D., Miller, V., and Mesbah, A. (2022). Performance-oriented model learning for control via multi-objective Bayesian optimization. *Computers & Chemical Engineering*, 162, 107770.

Marco, A., Berkenkamp, F., Hennig, P., Schoellig, A.P., Krause, A., Schaal, S., and Trimpe, S. (2017). Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with Bayesian optimization. In *IEEE International Conference on Robotics and Automation (ICRA)*, 1557–1563.

Marco, A., Hennig, P., Bohg, J., Schaal, S., and Trimpe, S. (2016). Automatic lqr tuning based on gaussian process global optimization. In *IEEE international conference on robotics and automation*, 270–277.

Mesbah, A., Wabersich, K.P., Schoellig, A.P., Zeilinger, M.N., Lucia, S., Badgwell, T.A., and Paulson, J.A. (2022). Fusion of machine learning and MPC under uncertainty: What advances are on the horizon? In *Proceedings of the American Control Conference*, 342–357. IEEE.

Müller, S., von Rohr, A., and Trimpe, S. (2021). Local policy search with Bayesian optimization. *Advances in Neural Information Processing Systems*, 34, 20708–20720.

Nguyen, Q., Wu, K., Gardner, J.R., and Garnett, R. (2022). Local Bayesian optimization via maximizing probability of descent. *arXiv preprint arXiv:2210.11662*.

Papini, M., Binaghi, D., Canonaco, G., Pirotta, M., and Restelli, M. (2018). Stochastic variance-reduced policy gradient. In *International Conference on Machine Learning*, 4026–4035. PMLR.

Paulson, J.A., Makrygiorgos, G., and Mesbah, A. (2022). Adversarially robust Bayesian optimization for efficient auto-tuning of generic control structures under uncertainty. *AIChE Journal*, 68(6), e17591.

Paulson, J.A. and Mesbah, A. (2020). Data-driven scenario optimization for automated controller tuning with probabilistic performance guarantees. *IEEE Control Systems Letters*, 5(4), 1477–1482.

Pautrat, R., Chatzilygeroudis, K., and Mouret, J.B. (2018). Bayesian optimization with automatic prior selection for data-efficient direct policy search. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 7571–7578. Brisbane.

Penubothula, S., Kamanchi, C., Bhatnagar, S., et al. (2021). Novel first order Bayesian optimization with an application to reinforcement learning. *Applied Intelligence*, 51(3), 1565–1579.

Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., and De Freitas, N. (2015). Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1), 148–175.

Shekhar, S. and Javidi, T. (2021). Significance of gradient information in Bayesian optimization. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2836–2844.

Sutton, R.S., McAllester, D., Singh, S., and Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems*, 12, 1057–1063.

Turchetta, M., Krause, A., and Trimpe, S. (2020). Robust model-free reinforcement learning with multi-objective Bayesian optimization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 10702–10708.

Williams, C.K. and Rasmussen, C.E. (2006). *Gaussian Processes for Machine Learning*. MIT Press, Cambridge.

Williams, R.J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3), 229–256.

Wu, J., Poloczek, M., Wilson, A.G., and Frazier, P. (2017). Bayesian optimization with gradients. *Advances in Neural Information Processing Systems*, 30.