Constrained Bayesian Optimization under Partial Observations: Balanced Improvements and Provable Convergence *

Shengbo Wang[1] and Ke Li[2]

[1]School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[2]Department of Computer Science, University of Exeter, EX4 4RN, Exeter, UK
shnbo.wang@foxmail.com, k.li@exeter.ac.uk

Abstract: The partially observable constrained optimization problems (POCOPs) impede data-driven optimization techniques since an infeasible solution of POCOPs can provide little information about the objective as well as the constraints. We endeavor to design an efficient and provable method for expensive POCOPs under the framework of constrained Bayesian optimization. Our method consists of two key components. Firstly, we present an improved design of the acquisition functions that introduce balanced exploration during optimization. We rigorously study the convergence properties of this design to demonstrate its effectiveness. Secondly, we propose Gaussian processes embedding different likelihoods as the surrogate model for partially observable constraints. This model leads to a more accurate representation of the feasible regions compared to traditional classification-based models. Our proposed method is empirically studied on both synthetic and real-world problems. The results demonstrate the competitiveness of our method for solving POCOPs.

Keywords: Bayesian optimization, constrained optimization, expected improvement, Gaussian processes, expectation propagation.

1 Introduction

The black-box constrained optimization problem considered in this paper is formulated as:

$$\text{minimize } f(\mathbf{x}) \quad \text{subject to} \quad \vec{g}(\mathbf{x}) \leq 0, \tag{1}$$

where $\mathbf{x} = (x_1, \cdots, x_n)^\top \in \Omega$ denotes the decision variable, $\Omega = [x_i^{\mathrm{L}}, x_i^{\mathrm{U}}]_{i=1}^n \subset \mathbb{R}^n$ denotes the search space, x_i^{L} and x_i^{U} are the lower and upper bounds of x_i respectively. The objective function $f(\mathbf{x})$ and m constraint functions $\vec{g}(\mathbf{x}) = (g_1(\mathbf{x}), \cdots, g_m(\mathbf{x}))^\top$ are: *i*) *analytically unknown*, i.e., we do not have access to f and $\vec{g}$ directly, but to $\mathbf{x}$ to be determined and their observations $f(\mathbf{x})$ and $\vec{g}(\mathbf{x})$ instead; *ii*) *computationally expensive*; and *iii*) *partially observable*, i.e., the values of f and $\vec{g}$ are not observable/measurable when $\mathbf{x}$ is infeasible. We denote the unknown feasible space by $\chi = \{\mathbf{x} \in \Omega | \vec{g}(\mathbf{x}) \leq 0\}$. Partially observable constrained optimization problems (POCOPs) are not uncommon in real-life applications. For example, a robot control task will be suspended when a collision or excessive instantaneous power consumption is detected, where the feedback is merely a failure message rather than any reward [1]. An AutoML task will be terminated without outputting the performance of a hyperparameter configuration but an error log when there is a memory overflow or computation timeout [2].

Bayesian optimization (BO) is recognized as an effective query-efficient framework for black-box optimization [3]. Although there have been dedicated efforts on constraint handling in the context of BO, a.k.a. constrained Bayesian optimization (CBO), most of them are however expected to work

with complete observations. Considering missing observations, existing CBO methods may become inefficient due to the following two issues.

- First, existing CBO methods risk overly exploiting known feasible regions in POCOPs. In particular, the update of an acquisition function such as expected improvement (EI) [4] stagnates when objective values are unobservable outside χ. This stagnation may cause cluttered observations in local feasible regions, resulting in an overconfidence effect in both surrogate modeling and candidate acquisition. Consequently, the efficiency of BO diminishes.
- Second, POCOPs generate mixed data from both feasible and infeasible solutions, which complicate sufficient exploitation using conventional surrogate models. When a probabilistic classifier like Gaussian process classifier (GPC) is employed to distinguish between feasible and infeasible solutions, the available observations lose their utility in refining the GPC model. Brockman et al. [5] leveraged value observations by using regression models and artificially injecting values into infeasible solutions. However, this approach potentially introduces erroneous priors, thereby compromising optimization efficacy.

Bearing these considerations in mind, this paper proposes a novel CBO framework for POCOPs. Our main contributions are outlined as follows.

- To address the risk of overly exploiting evaluated local feasible regions, we propose a novel acquisition function framework. It enhances EI with a balanced constraint handling technique, encapsulated in a general exploration function to enable global search during optimization. We theoretically analyze the convergence properties of this design, and further develop an instance of the exploration function, effectively enhancing the efficiency of our method for solving POCOPs.
- To fully leverage the mixed-observation characteristic of POCOPs, we propose heterogeneous-likelihood Gaussian processes (HLGPs), providing a promising representation of unknown constraints compared to classifier-based models. Further, we employ expectation propagation to manage the non-Gaussian inference, yielding an efficient Gaussian approximation of HLGPs.
- To demonstrate the efficacy of our proposed method, we perform a series of experiments on diverse benchmark problems. These include synthetic problems and real-world applications in reinforcement learning-based control design and machine learning hyperparameter optimization. Experimental results show the competitiveness and better efficiency of our method compared to selected state-of-the-art CBO methods.

2 Related Works

2.1 Constrained Bayesian Optimization

Classic CBO methods usually reshape an unconstrained acquisition function by incorporating feasibility considerations. A prominent approach, the EI with constraints (EIC), first introduced by Schonlau et al. [6], has been extensively employed to locate feasible solutions with high probability [7, 8]. To further enhance EIC's capability, Letham et al. [9] leveraged a quasi-Monte Carlo approximation regarding observation noises. Furthermore, various strategies such as integration [8], rollout [10], and a two-step lookahead algorithm [11], have been proposed to achieve better optimization efficiency. These approaches increase the exploration of unknown regions, albeit at the cost of computational complexity, hindering their scalability for high-dimensional problems.

From the perspective of uncertainty reduction, predictive entropy search (PESC), which selects feasible candidate solutions directly from the search space, offers attractive heuristics to constrained optimization problems [12]. However, the intractable nature of quadrature calculations during sampling in PESC has been a challenge. To address this, Takeno et al. [13] proposed the min-value entropy search, enabling the sampling process to operate more effectively within the objective space. This concept was subsequently extended to accommodate binary observations [2] and multi-objective

scenarios [14]. Besides, the fusion of EI and entropy search showed promise for enhancing exploration [15].

To harness the structure inherent in equation (1), researchers such as Gramacy et al. [16] and Picheny et al. [17] proposed the utilization of a Lagrangian method with slack variables, providing the capability to deal with equality constraints. Regarding problems marked by unknown constraints, Ariafar et al. [18] integrated BO with the alternating direction method of multipliers, thereby enabling the exploration of solutions even in the absence of feasible ones. Meanwhile, Eriksson and Poloczek [19] proposed the use of Thompson sampling combined with a trust region approach towards enhancement of the scalability, meanwhile incorporating a thoughtful design aimed at maintaining computational efficiency.

2.2 Surrogate Models for Constraints

The aforementioned CBO methods frequently employ Gaussian Process regression (GPR) and GPC to construct surrogate models for unknown constraints. Notably, classifiers such as GPC and support vector machine (SVM) have been found effective in sequential updates when the real value of an infringed constraint remains unobservable [2, 15, 18, 20, 21]. Yet, in partially observable scenarios, these classifiers exhibit limitations in utilizing available real-value observations, resulting in a dip in modeling performance. In response to this challenge, Marco et al. [1] enhanced the construction of GPR models by introducing a switched likelihood combined with mixed data observations. As an alternative solution, Pourmohamad and Lee [22] and Zhang et al. [23] proposed multivariate GPs (MVGP) with joint distributions of hybrid input to handle mixed observations.

2.3 Exploration for Unknown Feasible Regions

In light of the black-box nature of the problem at hand, discerning the feasibility of a solution becomes a significant concern. In this context, Parr et al. [24] interpreted the delicate balance between enhancing the probability of feasibility (POF) and optimizing the objective as a multi-objective optimization issue. Focusing on optimization, Picheny [25] developed a step-wise uncertainty reduction method, which capitalizes on the volume of feasible regions beneath the most promising solution observed up to that point, despite the method's considerable computational complexity. Furthermore, the EIC was adapted in [15,26] to deepen the understanding of global feasible regions. Regarding exploration, level-set or contour estimation techniques have been adapted to locate unknown feasible regions [27–29]. Yet, these methods can be excessively aggressive, thereby hindering optimization progress.

3 Preliminaries of CBO

Conventional BO starts from uniformly sampling a set of solutions according to a space-filling experimental design method. Thereafter, it sequentially updates its next sample until the given computational budget is exhausted. BO consists two main components: *i*) a surrogate model for approximating the true expensive objective function; and *ii*) an infill criterion (based on the optimization of an acquisition function) for deciding the next point of merit.

3.1 Surrogate Model

Given a set of training data $\mathcal{D} = \left\{\left(\mathbf{x}^i, f(\mathbf{x}^i)\right)\right\}_{i=1}^{N}$, we apply the GPR model to learn a Guassian process $\tilde{f}(\mathbf{x})$ with a prior mean function $m(\mathbf{x})$ and a noise-free likelihood [30]. For a candidate solution $\tilde{\mathbf{x}}$, the mean and variance of the target $f(\tilde{\mathbf{x}})$ can be predicted as:

$$\begin{aligned} \mu_f(\tilde{\mathbf{x}}) &= m(\tilde{\mathbf{x}}) + \mathbf{k}^{*\top} K^{-1} \mathbf{f}, \\ \sigma_f^2(\tilde{\mathbf{x}}) &= k(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}) - \mathbf{k}^{*\top} K^{-1} \mathbf{k}^*, \end{aligned} \tag{2}$$

Algorithm 1: Pseudo code of `CBOB`

Input: Initial dataset $\mathcal{D} = \left\{\left(\mathbf{x}^i, f(\mathbf{x}^i), \vec{g}(\mathbf{x}^i)\right)\right\}_{i=1}^{N_0}$, budget N, and priors of GPs
Output: The optimal feasible objective $f^*_{\mathcal{D}}$

for $k \leftarrow 1$ **to** N **do**
  Build a GPR model for the black-box objective;
  for $i \leftarrow 1$ **to** m **do**
    ▷ Update $\mathbf{g}_i$ in $\mathcal{D}$ with modified observations $\tilde{\mu}_g^i$ and $\tilde{\Sigma}_g^i$ using equation (13);
    ▷ Build an HLGP model based on $\mathbf{g}_i$;
  ▷ $\mathbf{x}^k \leftarrow \arg\max_{\tilde{\mathbf{x}}\in\Omega} \mathrm{EICB}(\tilde{\mathbf{x}}|\mathcal{D})$;
  if $\mathbf{x}^k$ *is feasible* **then**
    $\mathcal{D} \leftarrow \mathcal{D} \cup \left\{\left(\mathbf{x}^k, f(\mathbf{x}^k), \vec{g}(\mathbf{x}^k)\right)\right\}$;
  else
    ▷ $g_i^k \leftarrow +1$ for the i-th violated constraints;
    $\vec{g}(\mathbf{x}^k) = (g_1^k, \dots g_m^k)$;
    ▷ $\mathcal{D} \leftarrow \mathcal{D} \cup \left\{\left(\mathbf{x}^k, \mathrm{Null}, \vec{g}(\mathbf{x}^k)\right)\right\}$;

where $\mathbf{k}^*$ is the covariance matrix between X and $\tilde{\mathbf{x}}$, K is the covariance matrix of X, $X = \left(\mathbf{x}^1, \dots, \mathbf{x}^N\right)^\top$, and $\mathbf{f} = \left(f(\mathbf{x}^1) - m(\mathbf{x}^1), \dots, f(\mathbf{x}^N) - m(\mathbf{x}^N)\right)^\top$. In this paper, we use the Matérn 5/2 as the kernel function combined with the constant mean function for all GP models by default. As for the i-th constraint in (1), it will be modeled by an independent GP model $\tilde{g}_i(\tilde{\mathbf{x}})$ whose predictive mean and variance are denoted by $\mu_g^i(\tilde{\mathbf{x}})$ and $\sigma_g^i(\tilde{\mathbf{x}})$ respectively.

3.2 Infill Criterion

Instead of directly working on $\tilde{f}(\mathbf{x})$, the actual search process of BO is driven by an acquisition function that naturally strikes a balance between exploitation of the predicted optimum and exploration regarding uncertainty. This paper applies the widely used EI to serve this purpose:

$$\mathrm{EI}(\tilde{\mathbf{x}}|\mathcal{D}) = \sigma_f(\tilde{\mathbf{x}})\big(z\Phi_f(z) + \phi_f(z)\big), \tag{3}$$

where $z = \frac{f^*_{\mathcal{D}} - \mu_f(\tilde{\mathbf{x}})}{\sigma_f(\tilde{\mathbf{x}})}$, $f^*_{\mathcal{D}} = \min_{(\mathbf{x}, f(\mathbf{x}))\in\mathcal{D}} f(\mathbf{x})$, Φ_f and ϕ_f denote the cumulative distribution function and probability density function according to $\tilde{f}$, respectively.

To tackle unknown constraints, EIC was proposed as a product of the EI with POF [7]:

$$\mathrm{EIC}(\tilde{\mathbf{x}}|\mathcal{D}) = \mathrm{EI}(\tilde{\mathbf{x}}|\mathcal{D}) \cdot \mathrm{POF}(\tilde{\mathbf{x}}), \tag{4}$$

with

$$\mathrm{POF}(\tilde{\mathbf{x}}) = \mathbb{P}\left[\vec{g}(\tilde{\mathbf{x}}) \le \lambda\right] = \prod_{i=1}^{m} \Phi_g^i(\lambda), \tag{5}$$

where Φ_g^i denotes the cumulative distribution function of the i-th constraint based on a GPR model $\tilde{g}_i(\tilde{\mathbf{x}}) \sim \mathcal{N}(\mu_g^i(\tilde{\mathbf{x}}), {\sigma_g^i}^2(\tilde{\mathbf{x}}))$, λ is the threshold of a feasible level and is set to a constant 0 in this paper.

4 Proposed Method

This section delineates the implementation of our method, the CBO with balance (dubbed `CBOB`), for POCOPs. As shown in Algorithm 1, `CBOB` adheres to the conventional CBO procedure while introducing two unique algorithmic components (highlighted by ▷). The first is a framework for designing an acquisition function, facilitating balanced exploration by effectively harnessing the surrogate models of constraints. The second is a bespoke GP model, specifically formulated to model constraints using partial observations.

4.1 A Framework for Acquisition Function Design

Under the condition of partial observations, the EI function only updates upon evaluation of a feasible solution, while the POF predominantly targets known feasible regions. This results in an overemphasis on known feasible regions by the EIC, particularly when tackling POCOPs. Inspired by [25], we posit that prioritizing search towards less explored regions can enhance exploratory capability, thus promoting more global search behaviors in a CBO method. This approach has been empirically substantiated in [10, 11, 15]. In this work, we propose a dynamic version of POF (DPOF) that incorporates an additional exploration capability, rather than prioritizing the most uncertain region indiscriminately, as follows:

$$\mathrm{DPOF}(\tilde{\mathbf{x}}) = \prod_{i=1}^{m} \mathrm{Proj}_{[0,1]}\big[(\rho^i(\tilde{\mathbf{x}}) + 1)\Phi_g^i(\lambda)\big], \tag{6}$$

where Proj clips values outside $[0, 1]$ to the boundaries, and ρ^i denotes a general exploration function defined below.

Definition 1 (Exploration function)**.** *A smooth function $\rho^i(\mathbf{x}) : \Omega \to [0, \bar{\rho}]$ is a valid exploration function if it is bounded by $\bar{\rho} > 0$ and $\rho^i(\tilde{\mathbf{x}}) = 0, \forall \sigma_g^i(\mathbf{x}) = 0$.*

With an exploration function ρ^i, DPOF assigns more weights to unknown regions than POF to facilitate a global search. However, to prevent excessive exploration and maintain a high probability of obtaining feasible solutions, we multiply $\Phi_g^i(\lambda)$ with ρ^i in equation (6). Alternatively, this could be viewed as introducing a dynamic constraint threshold $\lambda(\tilde{\mathbf{x}}) = {\Phi_g^i}^{-1}(\mathrm{DPOF}^i(\tilde{\mathbf{x}}))$ that varies across different candidate solutions, where ${\Phi_g^i}^{-1}$ denotes the inverse function of Φ_g^i and $\mathrm{DPOF}^i(\tilde{\mathbf{x}})$ is the i-th factor in equation (6). Note that when $\rho^i \equiv 0$, DPOF simplifies to the traditional POF. Building on this, we introduce a new acquisition function, termed as EI with constraint and balance (EICB), formulated as a product of EI and DPOF:

$$\mathrm{EICB}(\tilde{\mathbf{x}}|\mathcal{D}) = \mathrm{EI}(\tilde{\mathbf{x}}|\mathcal{D}) \cdot \mathrm{DPOF}(\tilde{\mathbf{x}}). \tag{7}$$

Theorem 1. *Assume that the constraint values are fully observable. Assume also that the involved GPs are non-degenerate and satisfy the no-empty-ball property [31]. Let $\mathcal{D}$ be the collected observations with $\big(\mathbf{x}^1, f(\mathbf{x}^1)\big)$ fixed in χ while $\big\{\big(\mathbf{x}^i, f(\mathbf{x}^i)\big)\big\}_{i=2}^{N}$ are sequentially chosen by*

$$\mathbf{x}^i = \arg\max_{\tilde{\mathbf{x}} \in \Omega} \mathrm{EICB}(\tilde{\mathbf{x}}|\mathcal{D}). \tag{8}$$

Then, as $N \to \infty$, almost surely:

1. *the acquisition function $\sup_{\tilde{\mathbf{x}} \in \Omega} \mathrm{EICB}(\tilde{\mathbf{x}}|\mathcal{D}) \to 0$;*
2. *the evaluated best objective $f_{\mathcal{D}}^* \to f_{\chi}^*$;*

where f_{χ}^* represents the global optimum of problem (1).

The proof of Theorem 1 is sketched in Section A of the supplementary document. This theorem suggests that the incorporation of ρ^i as designed in equation (6) does not undermine the asymptotic convergence capability of EI-based acquisition functions, such as EICB. In the following subsection, we propose an instance of the exploration function under Definition 1 that outperforms the EIC in terms of efficiently conducting global optimization for POCOPs.

4.1.1 An instance of the exploration function

In the context of EICB framework, exploration during optimization can be facilitated by an apt design of ρ^i. In this paper, we concentrate on identifying promising constraint boundaries, as opposed to aggressively targeting the most uncertain regions, a tactic often employed in level-set estimation and

active learning [27–29, 32]. To this end, we first define a utility function representing the potential of being the boundary (POB) for the i-th constraint at $\tilde{\mathbf{x}} \in \Omega$ as follows:

$$\mathrm{POB}^i(\tilde{\mathbf{x}}) = \begin{cases} 1, & \tilde{g}_i(\tilde{\mathbf{x}}) \in [-\varepsilon(\tilde{\mathbf{x}}),\ \varepsilon(\tilde{\mathbf{x}})]\,, \\ 0, & \text{otherwise}, \end{cases} \tag{9}$$

where $\varepsilon(\tilde{\mathbf{x}}) = \beta\sigma_g^i(\tilde{\mathbf{x}})$ and $\beta > 0$ represents a confidence level. Taking the expectation of equation (9) over the predicted distribution of $\tilde{g}_i(\tilde{\mathbf{x}})$ and defining $\bar{g}_i(\tilde{\mathbf{x}}) = \mu_g^i(\tilde{\mathbf{x}})/\sigma_g^i(\tilde{\mathbf{x}})$, we obtain a valid exploration function as:

$$\rho^i(\tilde{\mathbf{x}}) = \Phi\left(\beta - \bar{g}_i(\tilde{\mathbf{x}})\right) - \Phi\left(-\beta - \bar{g}_i(\tilde{\mathbf{x}})\right), \tag{10}$$

where Φ is the cumulative distribution function of $\mathcal{N}(0, 1)$.

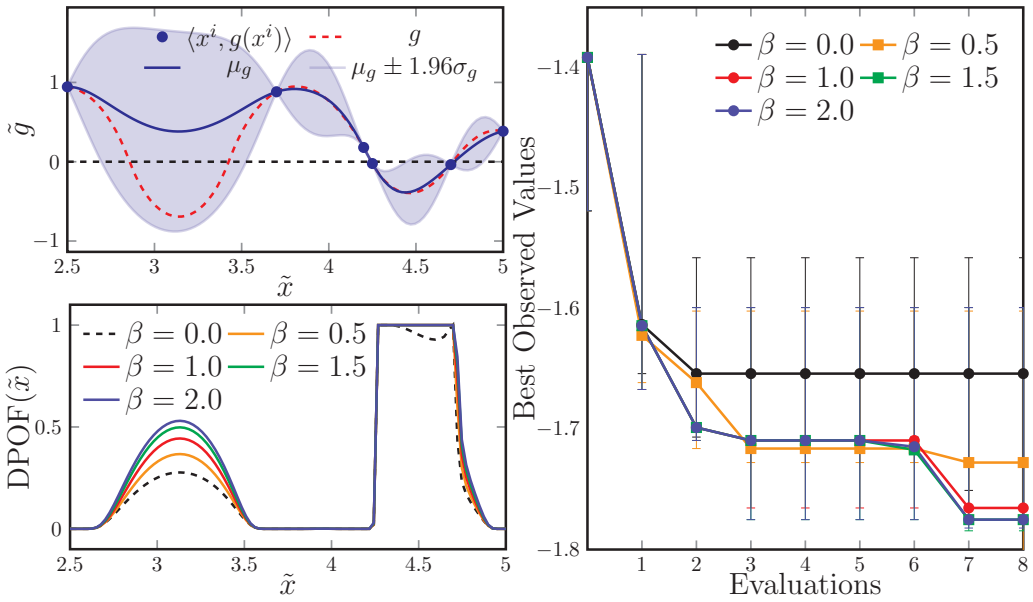

Figure 1: Illustration of EICB with equation (10) by a toy 1-D example (please refer to Section B.1 of the supplementary document for more details). (Left up) The constraint surrogate model with 6 observed data pairs where the red dotted line denotes the true function. (Left down) The DPOF with different β on $[2.5, 5]$, where the dotted line with $\beta = 0$ reduces to POF. (Right) Optimization trajectories on $[0, 10]$ with 5 randomly repeated experiments.

The illustrative example in Fig. 1 demonstrates how DPOF, given a surrogate model for a constraint, assigns more weight to the unknown feasible region ($[2.85, 3.45]$ in this case) as β increases. For a previously located feasible region such as $[4.3, 4.7]$, DPOF provides equal weights (approximately 1) to all candidates within this region when $\beta \geq 0.5$. In contrast, POF ($\beta = 0$) assigns differentiated weights based on different $\Phi_g(0)$ values. By refining the boundary, DPOF ensures a more balanced weight distribution within the located feasible region, hence the nomenclature, EICB. We posit that this balanced approach enhances the decision-making capabilities of EI, as compared to the imbalanced weights scenario posed by POF. As a positive consequence, EICB encourages greater exploration towards unknown regions. As ρ^i in equation (10) is bound by 1, DPOF can assign a maximum of $2\Phi_g(0)$ to any given candidate. Empirically, this subtle adjustment leads to enhanced optimization efficiency as evidenced in Fig. 1, thanks to the introduction of exploration. Besides, we provide a more aggressive design of ρ^i that bolsters the reduction of uncertainty in global feasible regions. Due to the space restriction, this discussion is presented in Section B.2 of the supplementary document.

4.2 Surrogate Models Under Partial Observations

When dealing with partially observable constraints, observations are composed of two distinct aspects: *i*) the actual values associated with feasible solutions, and *ii*) the truncated distribution of possible

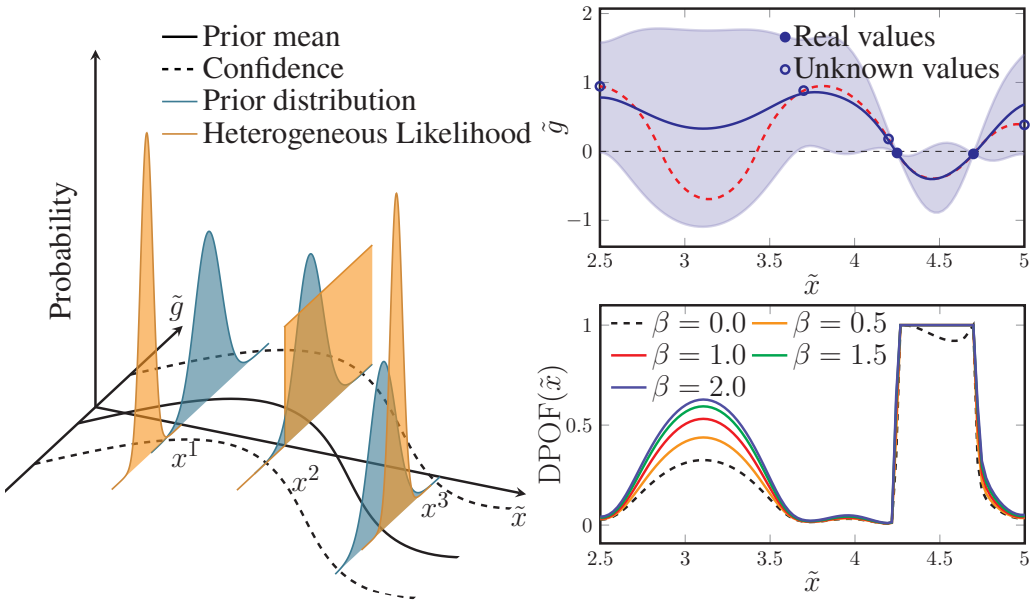

Figure 2: Illustration of HLGPs. (Left) Three observations with heterogeneous likelihood distributions, including a truncated distribution on x^2, and Gaussian distributions on x^1 and x^3. (Right up) An EP-based HLGP model with partial observations. (Right down) The DPOF with different β.

values for all solutions, such as $\mathbb{P}(\vec{g} > 0) = 1$, as depicted in Fig. 2. The simultaneous consideration of these two types of observations can be achieved by attaching individual likelihood distributions to feasible/infeasible solutions, as is done in HLGP.

For the i-th constraint, the posterior of a latent function, $p(\tilde{\mathbf{g}}_i|X, \mathbf{g}_i)$, within an HLGP model is determined via the Bayes rule using the prior distribution $p(\tilde{\mathbf{g}}_i|X) = \mathcal{N}(\mathbf{0}, K)$ in equation (2), along with the individual likelihood distributions. Specifically, the likelihood can be expressed as:

$$p(g_i^k|\tilde{g}_i^k) = \begin{cases} \mathcal{N}\left(g_i(\mathbf{x}^k), \sigma^2\right), & \text{if } g_i(\mathbf{x}^k) \leq 0, \\ \Phi(\alpha^{-1} g_i(\mathbf{x}^k)), & \text{if } g_i(\mathbf{x}^k) > 0, \end{cases} \tag{11}$$

where $k \in \{1, \ldots, N\}$, $\mathbf{g}_i = (g_i^1, \ldots, g_i^N)^\top$ represents N observations of $g_i(X)$, $\tilde{\mathbf{g}}_i = (\tilde{g}_i^1, \ldots, \tilde{g}_i^N)^\top$ denotes N latent functions, $\sigma \geq 0$ stands for the noise level, and $\alpha > 0$ is a scaling parameter. We set $\sigma = 10^{-6}$ to indicate a noise-free environment and $\alpha = 10^{-6}$ to approximate the truncating step function that implies $\mathbb{P}(g_i(\mathbf{x}^k) > 0) = 1$, $\forall g_i(\mathbf{x}^k) > 0$ [33].

4.2.1 HLGP Inference via Expectation Propagation

In this paper, we employ expectation propagation (EP) [34], a principled and highly efficient approach to handle non-Gaussian likelihoods. This provides Gaussian approximations to both the posterior and predicted distributions of HLGP. First, the posterior is formulated as:

$$p(\tilde{\mathbf{g}}_i|X, \mathbf{g}_i) = \frac{1}{Z} p(\tilde{\mathbf{g}}_i|X) \prod_{k=1}^{N} p(g_i^k|\tilde{g}_i^k), \tag{12}$$

where Z is the normalization factor. For the k-th observation g_i^k, EP assigns it an un-normalized Gaussian distribution $t_i^k \triangleq \tilde{Z}_i^k \mathcal{N}\left(\tilde{\mu}_i^k, \tilde{\sigma}_i^{k\,2}\right)$ to locally approximate its exact likelihood. In this vein, the posterior is approximated by:

$$\begin{aligned} p(\tilde{\mathbf{g}}_i|X, \mathbf{g}_i) &\approx \frac{1}{Z_{\text{EP}}} p(\tilde{\mathbf{g}}_i|X) \prod_{k=1}^{N} t_i^k = \mathcal{N}(\boldsymbol{\mu}_g^i, \Sigma_g^i) \\ \text{with} \quad \boldsymbol{\mu}_g^i &= \Sigma_g^i \tilde{\Sigma}_g^{i\,-1} \tilde{\boldsymbol{\mu}}_g^i \quad \text{and} \quad \Sigma_g^i = (K + \tilde{\Sigma}_g^{i\,-1})^{-1}, \end{aligned} \tag{13}$$

where $\tilde{\boldsymbol{\mu}}_g^i = \left(\tilde{\mu}_i^1, \cdots, \tilde{\mu}_i^N\right)^\top$, $\tilde{\Sigma}_g^i$ denotes a diagonal matrix with the k-th element being $\tilde{\sigma}_i^{k\,2}$, and Z_{EP} is the marginal likelihood. The site parameters in t_i^k of a Gaussian likelihood in equation (11) are valued by $\tilde{Z}_i^k = 1$, $\tilde{\mu}_i^k = g_i(\mathbf{x}^k)$, $\tilde{\sigma}_i^k = \sigma$. Differently, the site parameters of a non-Gaussian likelihood in equation (11) should be computed by the moment matching [33]. Detailed formulations of this part are delineated in Section C of the supplementary document. For a candidate solution $\tilde{\mathbf{x}}$, the mean and variance of the HLGP model $\tilde{g}_i(\tilde{\mathbf{x}})$ are predicted as:

$$\begin{aligned} \mu_g^i(\tilde{\mathbf{x}}) &= m(\tilde{\mathbf{x}}) + \mathbf{k}^{*\top}(K + \tilde{\Sigma}_g^i)^{-1}\tilde{\boldsymbol{\mu}}_g^i, \\ \sigma_g^{i\,2}(\tilde{\mathbf{x}}) &= k(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}) - \mathbf{k}^{*\top}(K + \tilde{\Sigma}_g^i)^{-1}\mathbf{k}^*. \end{aligned} \tag{14}$$

In principle, the hyperparameters of EP-based GP models should be updated by maximizing the marginal likelihood Z_{EP}. Since (14) resembles (2), from another perspective, EP algorithm serves as a data generator for HLGPs, i.e., assigning *virtual observations* for infeasible solutions with an estimation of noise levels. Accordingly, the hyperparameters of an HLGP model can be optimized by maximizing the marginal likelihood of a vanilla GPR model using these injected observations rather than Z_{EP} for better computation efficiency, as noted in [30].

4.2.2 Comparison with GPC

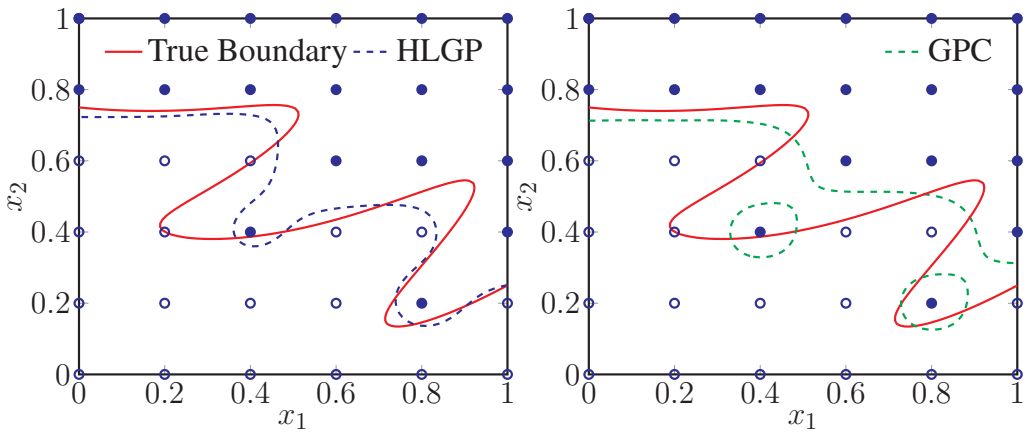

Figure 3: Different curves of feasible boundary predicted by (Left) an HLGP model versus (Right) a GPC model.

Finally, we present a brief comparison between the HLGP models in our proposed method and the GPC models frequently used in existing CBO algorithms for unobservable constraints. Notably, executing CBO routines is also feasible by only considering the truncated value distribution in equation (11). We contend that HLGP, by leveraging available observations, constructs a more reliable surrogate model than GPC. As illustrated in Fig. 3, 36 equidistant solutions are evaluated, with constraint values being unobservable for half. The true boundary and the predictions made by HLGP and GPC are provided for performance assessment. The GPC model delineates the boundary by maximizing the distance between the nearest distinct solutions, mirroring the approach of an SVM [30]. This results in the prediction of three disconnected feasible regions, deviating from the true feasible region. In contrast, HLGP predicts a connected feasible region that covers the majority of the true one. As such, we anticipate that HLGP, with its superior modeling capability, can enhance optimization for POCOPs.

4.3 Summary of the CBOB Algorithm

To improve the optimization efficiency for POCOPs, CBOB fully exploits available observations by re-designing a more balanced acquisition function and constructing principled surrogate models from mixed observations. Our method combines a boundary-based exploration function, as in equation (10),

and non-informative likelihoods as in equation (11). Moreover, CBOB preserves ample flexibility in designing exploration functions and likelihoods, thus making it adaptable to individual optimization problems and suitable for further investigations.

- Building on the concept of exploring potentially feasible regions as [15, 26], CBOB introduces extra exploration during optimization but with robust theoretical support underpinning this design. Moreover, our EICB maintains the computational efficiency of the original EIC, unlike the methods proposed in [8, 10, 11].
- Inspired by the level-set estimation methods [29], we propose an innovative exploration function as equation (10) that emphasizes potential boundaries, integrating with the POF for efficient optimization. This design avoids aggressively evaluating unknown regions, which is a common tactic in active learning [27, 28, 32].
- By employing HLGPs, we build a better surrogate model for each partially observable constraint function, outperforming GPC-based methods [2, 15, 18, 20, 21]. With the aid of EP and its generated virtual observations, we manage to construct HLGP models with commendable computational efficiency, resulting in an improvement over other models with mixed observations [22, 23].

5 Experiment Setup

In this section, we present the experimental settings used in our empirical study.

5.1 Benchmark Suite

Our experiments consider various optimization tasks, including synthetic problems, engineering design cases, hyperparameter optimization (HPO) problems based on scikit-learn [35], and reinforcement learning tasks based on Open AI Gym [5], to constitute our benchmark suite. In addition, we consider the following two scenarios of POCOPs.

- S1: The first scenario is that only f is partially observable. Specifically, problems include 10D Keane's bump function (KBF) [36], 4D welded beam design (WBD) [37], 7D HPO of XGBoost on the California housing dataset with a model size constraint (XGB-H), and 12D Lunar Landing with an energy constraint (Lunar) [38].
- S2: The second scenario considers both f and $\vec{g}$ are partially observable. Specifically, problems include 10D Ackley function with one constraint (Ackley) [1], 4D pressure vessel design (PVD) [39], 8D HPO of MLP on the digits dataset with a model size constraint (MLP-D), and 16D Swimmer with an energy constraint (Swimmer) [40].

5.2 Peer Algorithms

We consider three state-of-the-art CBO methods, including EIC [7] in the EI-based family, min-value entropy search with constraints (MESC) [13] in the information-theoretic family, and Thompson sampling with constraints (TSC) [19] in the stochastic sampling family. For S2, note that both EIC and MESC can handle binary constraint feedback [2, 20]. For S1, all algorithms use GPR to build the surrogate models. For S2, we choose either HLGP or GPC for modeling constraints. In particular, we use a dedicated subscript to represent the corresponding surrogate model, e.g., $\texttt{EIC}_c$ and $\texttt{EIC}_h$ denote EIC with GPC and HLGP, respectively.

5.3 General Settings

All algorithms are implemented according to their open-source code [13, 19]. In MESC, the optimal solutions are sampled 20 times. Both MESC and TSC sample with a grid size of 1 000. For CBOB with

Algorithm		KBF	WBD	XGB-H	Lunar
EIC	BOV	−0.33	2.47	0.281	255
	ROF	99.3%	75.3%	22.4%	82.3%
MESC	BOV	−0.29	2.28	**0.271**	255
	ROF	99.5%	18.3%	58.1%	84.5%
TSC	BOV	−0.26	2.47	0.283	247
	ROF	99.8%	77.9%	51.2%	87.7%
CBOB	BOV	**−0.39**	**2.16**	0.274	**260**
	ROF	99.2%	67.4%	25.8%	70.0%

Table 1: The BOV and ROF of different algorithms in S1.

equation (10), we fix $\beta = 1.96$ to obtain a 95% confidence level. As equation (10) is a conservative design for exploration, we omit the study on more conservative behaviors with smaller β. Each experiment is independently repeated 20 times with shared random seeds. For all tasks, the Sobol sequence is used to generate $11 \times n$ initial samples, then 100 function evaluations (FEs) are performed in each experiment. Detailed settings of all algorithms and benchmark problems are presented in Section D of the supplemental document. The source code of our project is available[1].

6 Experiment Results

The optimization trajectories of all experiments are given in Figures 4 and 5. In addition, the median best-evaluated values (BOVs) and average ratios of feasible evaluations (ROFs) of different algorithms are presented in Tables 1 and 2. We empirically study the efficacy of CBOB from three aspects: *i*) the *improvements* on CBOB for EI-based CBO methods and GPC-based models; *ii*) the *competitiveness* of CBOB with other peer algorithms; and *iii*) the *relationship* between efficiency and exploration ability of CBOB.

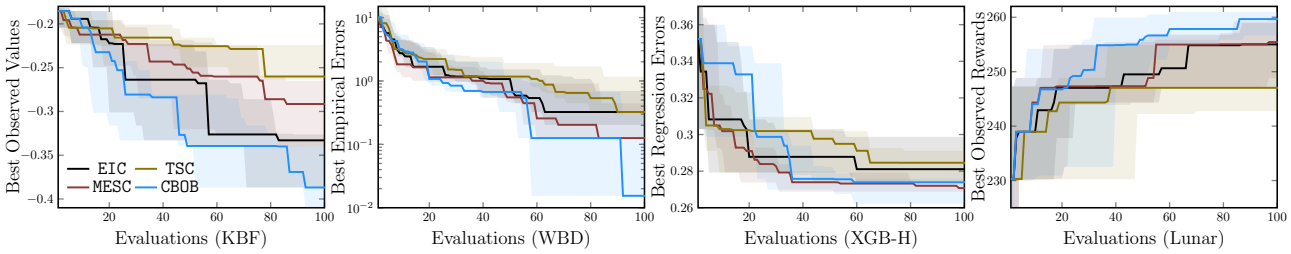

Figure 4: Optimization trajectories of different tasks in S1 (f is partially observable).

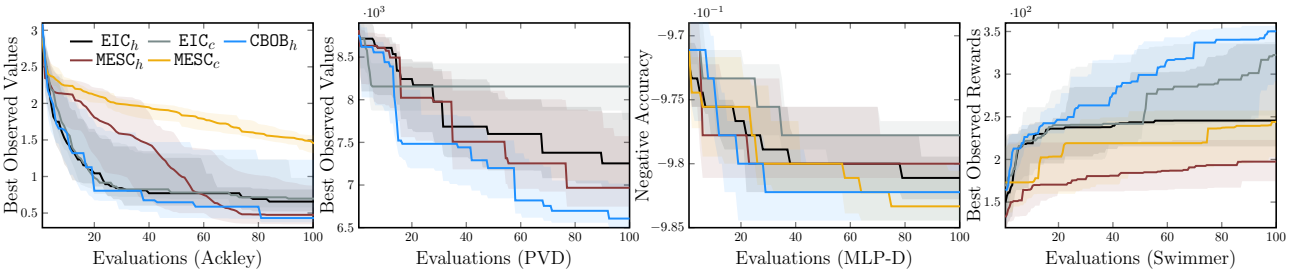

Figure 5: Optimization trajectories of different tasks in S2 (both f and $\vec{g}$ are partially observable).

[1] https://github.com/COLA-Laboratory/CBOB

Algorithm		Ackley	PVD	MLP-D	Swimmer
EIC$_c$	BOV	0.70	8155	0.978	322
	ROF	77.9%	5.12%	84.5%	82.3%
MESC$_c$	BOV	1.44	N/A	**0.983**	244
	ROF	78.2%	N/A	83.1%	83.5%
EIC$_h$	BOV	0.66	7598	0.981	245
	ROF	38.4%	3.13%	64.8%	67.2%
MESC$_h$	BOV	0.48	7507	0.98	197
	ROF	54.1%	3.78%	50.1%	79.8%
CBOB	BOV	**0.43**	**7198**	0.982	**350**
	ROF	38.0%	3.32%	65.5%	77.1%

Table 2: The BOV and ROF of different algorithms in S2.

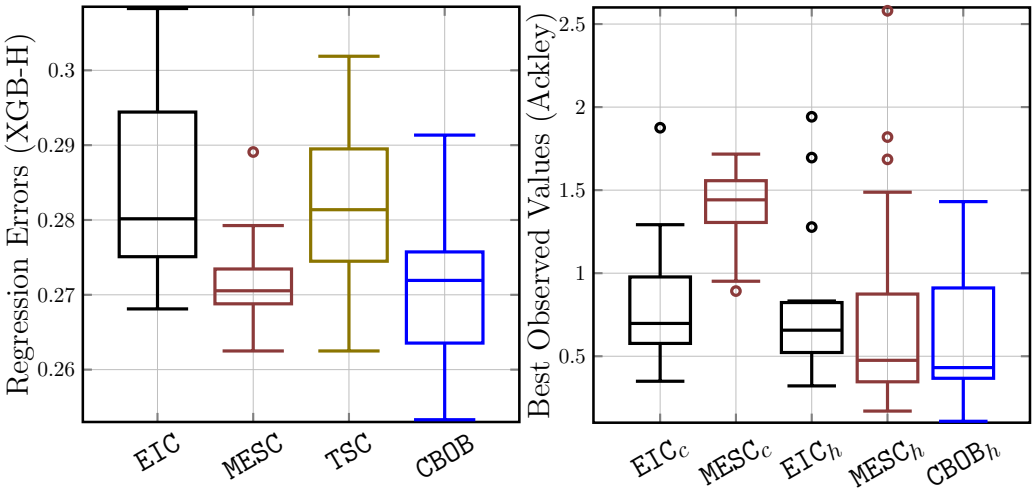

Figure 6: Box plot of the final best observed values of different algorithms on Ackley and XGB-H.

6.1 Improvements

Although EIC can be more efficient within 10 to 20 FEs, such as in WBD, XGB-H, Ackley, and MLP-D, EICB outperforms EIC in all experiments after 100 FEs, which demonstrates the design of DPOF. As the natural expense of exploration, the value deviation of EICB may be larger during the search. As given in Fig. 6, despite EIC$_c$ and EIC$_h$ have smaller deviations of the best evaluated values, EICB obtains a more promising result in a statistical sense. In addition, compared to GPC, HLGP does not always improve EIC and MESC, whereas benefiting CBOB well.

6.2 Competitiveness

The CBOB shows a strong competitiveness in all experiments against other CBO methods. In addition to the problems that EI-based methods perform well, such as KBF and Swimmer, CBOB remains competitive in problems that EI-based methods struggle, such as XGB-H, PVD and MLP-D. In comparison, MESC, as a promising CBO method in most problems, is inefficient in problems such as KBF and Swimmer. The TSC that showed high efficiency in fully observable environments with trust regions [19], struggles in solving POCOPs. Moreover, CBOB and other EI-based CBO methods show less efficiency in HPO problems, which agrees with the empirical results in [41].

6.3 Relationship between efficiency and exploration

In Tables 1 and 2, while CBOB obtains better feasible solutions, its ROFs are relatively low, i.e., more evaluated solutions of CBOB are infeasible. On the one hand, this agrees with the ideas of [11]

that exploration towards infeasible regions facilitates optimization efficiency of CBO methods. It also explains the larger deviation of CBOB in Fig. 6 that infeasible evaluations return little information in POCOPs. Besides, we find this exploration effective as the number of outliers of CBOB reduces, such as MESC_h and CBOB in Ackley of Fig. 6. On the other hand, since EI has already considered exploration, EIC can also have fewer ROFs during the search. Differently, in order not to break the balance of EI between exploration and exploitation, CBOB assigns more balanced weights for constraint handling. We highlight that this idea can also be integrated with other acquisition functions, such as the probability of improvement and parzen estimator [3].

7 Concluding Remarks

This paper designs CBOB that fully exploits the available observations for better exploration and surrogate modeling, by both theoretical and empirical analysis, demonstrating that CBOB has the potential to be a promising CBO method for POCOPs. Due to the space restriction, we provide some further discussions on the limitations and potential impact of this work in Section E of the supplementary document. Further investigations include the in-depth theoretical study of CBOB and the design of the exploration functions that are suitable for individual problems. We will also endeavor to propose a risk-aware improvement of EICB regarding robustness and the reduction of infeasible evaluations, contributing to more practical optimization scenarios.

Acknowledgment

This work was supported in part by the UKRI Future Leaders Fellowship under Grant MR/S017062/1 and MR/X011135/1; in part by NSFC under Grant 62376056 and 62076056; in part by the Royal Society under Grant IES/R2/212077; in part by the EPSRC under Grant 2404317; in part by the Kan Tong Po Fellowship (KTP\R1\231017); and in part by the Amazon Research Award and Alan Turing Fellowship.

References

[1] A. Marco, D. Baumann, M. Khadiv, P. Hennig, L. Righetti, and S. Trimpe, "Robot learning with crash constraints," *IEEE Robotics Autom. Lett.*, vol. 6, no. 2, pp. 1439–1446, 2021.

[2] V. Perrone, I. Shcherbatyi, R. Jenatton, C. Archambeau, and M. W. Seeger, "Constrained bayesian optimization with max-value entropy search," *CoRR*, vol. abs/1910.07003, 2019. [Online]. Available: http://arxiv.org/abs/1910.07003

[3] R. Garnett, *Bayesian Optimization*. Cambridge University Press, 2023.

[4] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *J. Glob. Optim.*, vol. 13, no. 4, pp. 455–492, 1998.

[5] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," 2016.

[6] M. Schonlau, W. J. Welch, and D. R. Jones, *Global Versus Local Search in Constrained Optimization of Computer Models*. Institute of Mathematical Statistics, 1 1998, vol. 34, pp. 11–25.

[7] J. R. Gardner, M. J. Kusner, Z. E. Xu, K. Q. Weinberger, and J. P. Cunningham, "Bayesian optimization with inequality constraints," in *ICML'14: Proc. of the 31th International Conference on Machine Learning*, vol. 32. JMLR.org, 2014, pp. 937–945.

[8] M. A. Gelbart, J. Snoek, and R. P. Adams, "Bayesian optimization with unknown constraints," in *UAI'14: Proc. of the 13th Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2014, pp. 250–259.

[9] B. Letham, B. Karrer, G. Ottoni, and E. Bakshy, "Constrained bayesian optimization with noisy experiments," *Bayesian Anal.*, vol. 14, no. 2, pp. 495 – 519, 2019.

[10] R. Lam and K. Willcox, "Lookahead bayesian optimization with inequality constraints," in *NeurIPS'17: Annual Conference on Neural Information Processing Systems*, 2017, pp. 1890–1900.

[11] Y. Zhang, X. Zhang, and P. I. Frazier, "Two-step lookahead bayesian optimization with inequality constraints," in *NeurIPS'21: Annual Conference on Neural Information Processing Systems*, 2021, pp. 12 563–12 575.

[12] J. M. Hernández-Lobato, M. A. Gelbart, M. W. Hoffman, R. P. Adams, and Z. Ghahramani, "Predictive entropy search for bayesian optimization with unknown constraints," in *ICML'15: Proc. of the 32nd International Conference on Machine Learning 2015*, vol. 37. JMLR.org, 2015, pp. 1699–1707.

[13] S. Takeno, T. Tamura, K. Shitara, and M. Karasuyama, "Sequential and parallel constrained max-value entropy search via information lower bound," in *ICML'22, Proc. of the 39th International Conference on Machine Learning*, vol. 162. PMLR, 2022, pp. 20 960–20 986.

[14] S. Belakaria, A. Deshwal, and J. R. Doppa, "Max-value entropy search for multi-objective bayesian optimization," in *NeurIPS'19: Annual Conference on Neural Information Processing Systems*, 2019, pp. 7823–7833.

[15] D. V. Lindberg and H. K. Lee, "Optimization under constraints by applying an asymmetric entropy measure," *J. Comput. Graph. Stat.*, vol. 24, no. 2, pp. 379–393, 2015.

[16] R. B. Gramacy, G. A. Gray, S. L. Digabel, H. K. H. Lee, P. Ranjan, G. N. Wells, and S. M. Wild, "Modeling an augmented lagrangian for black-box constrained optimization," *Technometrics*, vol. 58, no. 1, pp. 1–11, 2016.

[17] V. Picheny, R. B. Gramacy, S. M. Wild, and S. L. Digabel, "Bayesian optimization under mixed constraints with a slack-variable augmented lagrangian," in *NeurIPS'16: Annual Conference on Neural Information Processing Systems*, 2016, pp. 1435–1443.

[18] S. Ariafar, J. Coll-Font, D. H. Brooks, and J. G. Dy, "ADMMBO: bayesian optimization with unknown constraints using ADMM," *J. Mach. Learn. Res.*, vol. 20, pp. 123:1–123:26, 2019.

[19] D. Eriksson and M. Poloczek, "Scalable constrained bayesian optimization," in *AISTATS'21: Proc. of the 24th International Conference on Artificial Intelligence and Statistics*, vol. 130. PMLR, 2021, pp. 730–738.

[20] F. Bachoc, C. Helbert, and V. Picheny, "Gaussian process optimization with failures: Classification and convergence proof," *J. Glob. Optim.*, vol. 78, no. 3, pp. 483–506, 2020.

[21] A. Candelieri, "Sequential model based optimization of partially defined functions under unknown constraints," *J. Glob. Optim.*, vol. 79, no. 2, pp. 281–303, 2021.

[22] T. Pourmohamad and H. K. H. Lee, "Multivariate stochastic process models for correlated responses of mixed type," *Bayesian Anal.*, vol. 11, pp. 797 – 820, 2016.

[23] Y. Zhang, Z. Dai, and B. K. H. Low, "Bayesian optimization with binary auxiliary information," in *UAI'19: Proc. of the 35th Conference on Uncertainty in Artificial Intelligence*, vol. 115. AUAI Press, 2019, pp. 1222–1232.

[24] J. M. Parr, A. J. Keane, A. I. Forrester, and C. M. Holden, "Infill sampling criteria for surrogate-based optimization with constraint handling," *Eng. Optim.*, vol. 44, no. 10, pp. 1147–1166, 2012.

[25] V. Picheny, "A stepwise uncertainty reduction approach to constrained global optimization," in *AISTATS'14: Proc. of the 17th International Conference on Artificial Intelligence and Statistics*, vol. 33. JMLR.org, 2014, pp. 787–795.

[26] Z. Wang and M. Ierapetritou, "Constrained optimization of black-box stochastic systems using a novel feasibility enhanced kriging-based method," *Comput. Chem. Eng.*, vol. 118, pp. 210–223, 2018.

[27] P. Ranjan, D. Bingham, and G. Michailidis, "Sequential experiment design for contour estimation from complex computer codes," *Technometrics*, vol. 50, no. 4, pp. 527–541, 2008.

[28] J. Bect, D. Ginsbourger, L. Li, V. Picheny, and E. Vázquez, "Sequential design of computer experiments for the estimation of a probability of failure," *Stat. Comput.*, vol. 22, no. 3, pp. 773–793, 2012.

[29] F. Bachoc, T. Cesari, and S. Gerchinovitz, "The sample complexity of level set approximation," in *AISTATS'21: Proc. of the 24th International Conference on Artificial Intelligence and Statistics*, vol. 130. PMLR, 2021, pp. 424–432.

[30] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 11 2005.

[31] E. Vazquez and J. Bect, "Convergence properties of the expected improvement algorithm with fixed mean and covariance functions," *J. Stat. Plan. Inference*, vol. 140, no. 11, pp. 3088–3095, 2010.

[32] B. J. Bichon, M. S. Eldred, L. P. Swiler, S. Mahadevan, and J. M. McFarland, "Efficient global reliability analysis for nonlinear implicit performance functions," *AIAA J.*, vol. 46, no. 10, pp. 2459–2468, 2008.

[33] J. Riihimäki and A. Vehtari, "Gaussian processes with monotonicity information," in *AISTATS'10: Proc. of the 13th International Conference on Artificial Intelligence and Statistics*. JMLR.org, 2010, pp. 645–652.

[34] T. P. Minka, "Expectation propagation for approximate bayesian inference," in *UAI '01: Proc. of the 17th Conference in Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 2001, pp. 362–369.

[35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. VanderPlas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[36] A. J. Keane, "Experiences with optimizers in structural design," in *Proc. of the conference on adaptive computing in engineering design and control*, vol. 94, 1994, pp. 14–27.

[37] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2, pp. 311–338, 2000.

[38] D. Eriksson, M. Pearce, J. R. Gardner, R. Turner, and M. Poloczek, "Scalable global optimization via local bayesian optimization," in *NeurIPS'19: Annual Conference on Neural Information Processing Systems*, 2019, pp. 5497–5508.

[39] C. A. C. Coello and E. Mezura-Montes, "Constraint-handling in genetic algorithms through the use of dominance-based tournament selection," *Adv. Eng. Informatics*, vol. 16, no. 3, pp. 193–203, 2002.

[40] L. Wang, R. Fonseca, and Y. Tian, "Learning search space partition for black-box optimization using monte carlo tree search," in *NeurIPS'20: Annual Conference on Neural Information Processing Systems*, 2020.

[41] S. Watanabe and F. Hutter, "c-tpe: Tree-structured parzen estimator with inequality constraints for expensive hyperparameter optimization," in *IJCAI'23: Proc. of the 32nd International Joint Conference on Artificial Intelligence*, 2023, pp. 4371–4379.

[42] J. Bect, F. Bachoc, and D. Ginsbourger, "A supermartingale approach to gaussian process based sequential design of experiments," *Bernoulli*, vol. 25, no. 4A, pp. 2883–2919, 2019.

[43] D. Ginsbourger, O. Roustant, and N. Durrande, "On degeneracy and invariances of random fields paths with applications in gaussian process modelling," *J. Stat. Plan. Inference*, vol. 170, pp. 117–128, 2016.

[44] A. G. de G. Matthews, M. van der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagrá, Z. Ghahramani, and J. Hensman, "Gpflow: A gaussian process library using tensorflow," *J. Mach. Learn. Res.*, vol. 18, pp. 40:1–40:6, 2017.

[45] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *IROS'12: IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.

Supplementary document: Technical Appendices

This document includes the detailed derivation, algorithm design, and experiment settings in the paper. Specifically, we present the theoretical analysis of the EICB framework in Section A. The empirical study of EICB, including another design of the exploration function and comparative experiments of EI-based CBO methods, is given in Section B. In Section C, we show a detailed derivation of the EP approximation for HLGP models. The experiment settings are presented in Section D. Finally in Section E, we discuss the limitations and potential impact of our work.

Section A. Theoretical analysis of EICB

We justify the design of EICB by studying its convergence property under the framework of the *consistency* research for EI [42] and vanilla EIC [20]. In the context of sequential design, let $\mathcal{F}_N$ denote the σ-algebra generated by the random variables $\mathbf{x}^1, Z^1, \ldots, \mathbf{x}^N, Z^N$ where Z^i is the union observation of $\left(f(\mathbf{x}^i), \vec{g}(\mathbf{x}^i)\right)$. Additionally, let $\mathcal{F}_{N,\tilde{\mathbf{x}}}$ be the σ-algebra generated by $\mathbf{x}^1, Z^1, \ldots, \mathbf{x}^N, Z^N$, $\tilde{\mathbf{x}}, \tilde{Z}$ with $\tilde{Z}$ the union observation of $(f(\tilde{\mathbf{x}}), \vec{g}(\tilde{\mathbf{x}}))$. Then, a sequential design strategy derived from the EICB acquisition framework takes the following form:

$$\mathbf{x}_{N+1} = \arg\max_{\tilde{\mathbf{x}}\in\Omega} \mathbb{E}_N \left[M_N^0 - M_{N,\tilde{\mathbf{x}}}^{\rho} \right], \tag{A.1}$$

in which

$$M_{N,\tilde{\mathbf{x}}}^{\rho} = \min_{\mathbf{x}\in\Omega,\, \mathbb{P}(\vec{g}(\mathbf{x})<\lambda(\mathbf{x})|\mathcal{F}_{N,\tilde{\mathbf{x}}})=1,\, \sigma_f(\mathbf{x}|\mathcal{F}_{N,\tilde{\mathbf{x}}})=0} \tilde{f}(\mathbf{x}), \tag{A.2}$$

$$M_N^0 = \min_{\mathbf{x}\in\Omega,\, \mathbb{P}(\tilde{g}(\mathbf{x})<0|\mathcal{F})=1,\, \sigma_f(\mathbf{x}|\mathcal{F}_N)=0} \tilde{f}(\mathbf{x}). \tag{A.3}$$

Definition 2 (Non-degenerate GPs [43])**.** *A non-degenerate GP model $\tilde{f}(\mathbf{x})$ predicts $\sigma_f(\mathbf{x}) = 0$ if and only if $(\mathbf{x}, f(\mathbf{x})) \in \mathcal{D}$.*

Definition 3 (No-empty-ball [31])**.** *Let $(\mathbf{x}^N)_{N\geq 1}$ be any sequence in Ω, and z be any solution in Ω. A GP model has the no-empty-ball (NEB) property, if its prediction error at z goes to zero, $\forall \epsilon > 0$, there always exists $N \geq 1$ such that $|z - \mathbf{x}^N| < \epsilon$.*

Proof of Theorem 1. When GPs are non-degenerate, equation (A.1) becomes equivalent to equation (8). Specifically, M_N^0 will be equal to $f_{\mathcal{D}}^*$ in equation (3), while $M_{N,\tilde{\mathbf{x}}}^{\rho}$ will be the predicted feasible objective value under dynamic constraint threshold implicitly defined by equation (6). Next, we present the criteria for *asymptotic convergence* of EICB.

The proof of the first statement, i.e., the convergence of EICB, consists of three steps.

Step 1. EICB serves as a stepwise uncertainty reduction (SUR) sequential design. For $N \geq 2$, a minimization version of equation (A.1) can be given as

$$\mathbf{x}_{N+1} = \arg\min_{\tilde{\mathbf{x}}\in\Omega} \mathbb{E}_N \left[H_{N,\tilde{\mathbf{x}}} \right], \tag{A.4}$$

in which

$$\begin{aligned} H_{N,\tilde{\mathbf{x}}} =& M_{N,\tilde{\mathbf{x}}}^{\rho} - M_N^0 \\ =& \mathbb{E}_{N,\tilde{\mathbf{x}}} \left[M_{N,\tilde{\mathbf{x}}}^{\rho} - \min_{\mathbf{x}\in\Omega,\, \vec{g}(\mathbf{x})<0} \tilde{f}(\mathbf{x}) \right]. \end{aligned} \tag{A.5}$$

The above equation holds since: *i)* M_N^0 is independent from $\tilde{\mathbf{x}}$, and *ii)* $\mathbb{E}_{N,\tilde{\mathbf{x}}}\left[M_{N,\tilde{\mathbf{x}}}^{\rho}\right] = M_{N,\tilde{\mathbf{x}}}^{\rho}$ for minimum operation. Therefore EICB strategy can be transformed into an equivalent SUR sequential design strategy for $H_{N,\tilde{\mathbf{x}}}$. Likewise, we define

$$H_N = \mathbb{E}_N \left[M_N^{\rho} - \min_{\mathbf{x}\in\Omega,\, \vec{g}(\mathbf{x})<0} \tilde{f}(\mathbf{x}) \right]. \tag{A.6}$$

Step 2. (H_N) **is a supermartingale.** For well-structured GP models and well-defined smooth functions ρ^i, we have: *i)* $\sigma_f(\mathbf{x}|\mathcal{F}_{N+1}) \leq \sigma_f(\mathbf{x}|\mathcal{F}_N)$ (based on the definition of GP predicted variance), and *ii)* $\mathbb{P}(\vec{g}(\mathbf{x}) < \lambda(\mathbf{x})|\mathcal{F}_N) = 1$ is sufficient for $\mathbb{P}(\vec{g}(\mathbf{x}) < \lambda(\mathbf{x})|\mathcal{F}_{N+1}) = 1$ (based on the non-increasing property of ρ^i on an evaluated solution $\mathbf{x}^N$). Therefore, the following inequality holds:

$$H_N - \mathbb{E}_N[H_{N+1}] = \mathbb{E}_N\left[M_N^\rho - M_{N+1}^\rho\right] \geq 0, \tag{A.7}$$

which implies that $(H_N)_{N\in\mathbb{N}}$ is a supermartingale. According to [20], there is $H_N - \mathbb{E}_N[H_{N+1}] \to 0$ as $N \to \infty$, and also

$$\sup_{\tilde{\mathbf{x}}\in\Omega}\left[H_N - \mathbb{E}_N[H_{N,\tilde{\mathbf{x}}}]\right] \to 0. \tag{A.8}$$

Step 3. $\mathrm{EICB}(\tilde{\mathbf{x}}|\mathcal{D})$ **converges to** 0 **almost surely.** Note that for N observed solutions, $M_N^\rho = M_N^0$ as $\lambda = 0$. According to equation (A.7), we also have

$$\begin{aligned}\sup_{\tilde{\mathbf{x}}\in\Omega}\mathbb{E}_N\left[M_N^\rho - M_{N,\tilde{\mathbf{x}}}^\rho\right] &\geq \sup_{\tilde{\mathbf{x}}\in\Omega}\mathbb{E}_N\left[M_N^0 - M_{N,\tilde{\mathbf{x}}}^\rho\right] \\ &\geq \sup_{\tilde{\mathbf{x}}\in\Omega}\mathrm{EI}(\tilde{\mathbf{x}}|\mathcal{D})\cdot\mathrm{DPOF}(\tilde{\mathbf{x}}).\end{aligned} \tag{A.9}$$

Therefore, with the same proof as that of Proposition 2.9 [42], for $N \to \infty$, (A.8) and (A.9) yield $\sup_{\tilde{\mathbf{x}}\in\Omega}\mathrm{EICB}(\tilde{\mathbf{x}}|\mathcal{D}) \to 0$. From equation (3), it can be further obtained that $\mathrm{DPOF}(\tilde{\mathbf{x}})\sigma_f(\tilde{\mathbf{x}}) \to 0$. This completes the proof for the first statement.

The second statement is proven by virtue of the global search ability of EI and corresponding dense evaluated solutions in χ if the NEB property is met. We complete the proof by providing the following facts: *i)* $\mathrm{DPOF}(\tilde{\mathbf{x}})\sigma_f(\tilde{\mathbf{x}}) \to 0$ holds from the first statement; *ii)* $\mathrm{DPOF}(\tilde{\mathbf{x}}) \geq \mathrm{POF}(\tilde{\mathbf{x}})$ since $\rho^i \geq 0$; *iii)* the variance of $\mathrm{POF}(\tilde{\mathbf{x}})$ will not go to zero according to its definition [20]; and *iv)* $\sigma_f(z|\mathcal{F}_N) \to 0$ for all sequences accordingly. Based on these facts, the sequence is almost surely dense in χ if the GP models have the NEB property according to Definition 3. As a result, $f_{\mathcal{D}}^*$ from any sequence converges to f_χ^* almost surely when $N \to \infty$. ■

Section B. Empirical analysis of EICB

B.1 The illustrative example in Fig. 1

The synthetic problem considered in Fig. 1 is given as:

$$\begin{aligned}\text{minimize} \quad & \cos(5x) - \sin(x)\sin(2x), \\ \text{subject to} \quad & \cos(5x) - \sin(x)\sin(2x) \leq 0,\end{aligned} \tag{B.1}$$

where the objective function and constraint function share the same analytical format. The total search space is $\Omega = [0, 10]$, while in the left of Fig. 1 only a sub-region ($[2.5, 5.0]$) is plotted for brevity. The initial evaluated points in the left of Fig. 1 include: 2 feasible points at $[4.25, 4.7]$, and 4 infeasible points $[2.5, 3.7, 4.2, 5.0]$. We further assume that both objective and constraint are fully observable to efficiently reveal the difference between EIC and EICB, while in all other examples and experiments, we consider POCOPs. The GPR models with Matérn 5/2 kernel and constant mean function are used to build the surrogate models.

In the right of Fig. 1, we conduct 5 repeated experiments in which 10 solutions are uniformly sampled for initialization. We use consistent random seeds across different acquisition functions to obtain the same initialization. The total budget for optimization is 8. The median, 1/4, and 3/4 quantiles of the best observed objective values are plotted.

B.2 Another instance of the exploration function

The presented design of an exploration function in equation (9) and equation (10) is relatively conservative compared to the level-set estimation and active learning techniques since it only considers

boundaries with high probability. We are going to introduce another exploration function that aims for uncertainty reduction of the global feasible regions. Inspired by [32], this can be achieved by modifying the utility function in equation (9) into

$$\begin{aligned}\mathrm{MUB}^i(\tilde{\mathbf{x}}) =& \max\{\varepsilon(\tilde{\mathbf{x}}) - |\tilde{g}_i(\tilde{\mathbf{x}})|, 0\} \\ =& \begin{cases} \varepsilon(\tilde{\mathbf{x}}) - \tilde{g}_i(\tilde{\mathbf{x}}), & \tilde{g}_i(\tilde{\mathbf{x}}) \in [0,\ \varepsilon(\tilde{\mathbf{x}})]\,, \\ \varepsilon(\tilde{\mathbf{x}}) + \tilde{g}_i(\tilde{\mathbf{x}}), & \tilde{g}_i(\tilde{\mathbf{x}}) \in [-\varepsilon(\tilde{\mathbf{x}}),\ 0)\,, \\ 0, & \text{otherwise.} \end{cases}\end{aligned} \tag{B.2}$$

We name it the most uncertain boundary (MUB) to indicate that it facilitates a reduction of the uncertainty of global feasible regions more efficiently.

Lemma 1. *Given the utility function in equation (B.2), the expectation of MUB (EMUB) over the predicted distribution of $\tilde{g}_i(\tilde{\mathbf{x}})$ takes the following form as*

$$\begin{aligned}\mathrm{EMUB}^i(\tilde{\mathbf{x}}) =& \varepsilon(\tilde{\mathbf{x}})\Big(\Phi\left(g_i^+(\tilde{\mathbf{x}})\right) - \Phi\left(g_i^-(\tilde{\mathbf{x}})\right)\Big) \\ & - \sigma_g^i(\tilde{\mathbf{x}})\Big(2\phi\left(-\bar{g}_i(\tilde{\mathbf{x}})\right) \\ & \qquad - \phi\left(g_i^+(\tilde{\mathbf{x}})\right) - \phi\left(g_i^-(\tilde{\mathbf{x}})\right)\Big) \\ & + \mu_g^i(\tilde{\mathbf{x}})\Big(2\Phi\left(-\bar{g}_i(\tilde{\mathbf{x}})\right) \\ & \qquad - \Phi\left(g_i^+(\tilde{\mathbf{x}})\right) - \Phi\left(g_i^-(\tilde{\mathbf{x}})\right)\Big),\end{aligned} \tag{B.3}$$

where $\bar{g}_i(\tilde{\mathbf{x}}) = \mu_g^i(\tilde{\mathbf{x}})/\sigma_g^i(\tilde{\mathbf{x}})$, $g_i^+(\tilde{\mathbf{x}}) = \beta - \mu_g^i(\tilde{\mathbf{x}})/\sigma_g^i(\tilde{\mathbf{x}})$, $g_i^-(\tilde{\mathbf{x}}) = -\beta - \mu_g^i(\tilde{\mathbf{x}})/\sigma_g^i(\tilde{\mathbf{x}})$, Φ and ϕ denote the cumulative distribution function and probability density function of $\mathcal{N}(0,1)$, respectively.

Proof. Let $\tilde{g}_i(\tilde{\mathbf{x}}) = \mu_g^i(\tilde{\mathbf{x}}) + \sigma_g^i(\tilde{\mathbf{x}})\epsilon$ with $\epsilon \sim \mathcal{N}(0,1)$. The expectation of the utility function of MUB is derived by

$$\begin{aligned}& \mathbb{E}_{\tilde{g}_i(\tilde{\mathbf{x}})\sim\mathcal{N}(\mu_g^i(\tilde{\mathbf{x}}),\sigma_g^i(\tilde{\mathbf{x}}))}\left[\mathrm{MUB}^i(\tilde{\mathbf{x}})\right] \\ =& \mathbb{E}_{\epsilon\sim\mathcal{N}(0,1)}\left[\mathrm{MUB}^i(\tilde{\mathbf{x}})\right] \\ =& \int_{\varepsilon\geq|\mu_g^i(\tilde{\mathbf{x}})+\sigma_g^i(\tilde{\mathbf{x}})\epsilon|} \left(\varepsilon - |\mu_g^i(\tilde{\mathbf{x}}) + \sigma_g^i(\tilde{\mathbf{x}})\epsilon|\right)\phi\left(\epsilon\right)d\epsilon \\ =& \int_{\bar{g}_i(\tilde{\mathbf{x}})}^{g_i^+(\tilde{\mathbf{x}})} \left(\varepsilon - \mu_g^i(\tilde{\mathbf{x}}) - \sigma_g^i(\tilde{\mathbf{x}})\epsilon\right)\phi\left(\epsilon\right)d\epsilon \\ & + \int_{g_i^-(\tilde{\mathbf{x}})}^{\bar{g}_i(\tilde{\mathbf{x}})} \left(\varepsilon + \mu_g^i(\tilde{\mathbf{x}}) + \sigma_g^i(\tilde{\mathbf{x}})\epsilon\right)\phi\left(\epsilon\right)d\epsilon \\ =& \varepsilon\left(\Phi\left(g_i^+(\tilde{\mathbf{x}})\right) - \Phi\left(g_i^-(\tilde{\mathbf{x}})\right)\right) \\ & + \mu_g^i(\tilde{\mathbf{x}})\left(2\Phi(\bar{g}_i(\tilde{\mathbf{x}})) - \Phi(g_i^+(\tilde{\mathbf{x}})) - \Phi(g_i^-(\tilde{\mathbf{x}}))\right) \\ & + \sigma_g^i(\tilde{\mathbf{x}})\left(\int_{g_i^-(\tilde{\mathbf{x}})}^{\bar{g}_i(\tilde{\mathbf{x}})} \epsilon\phi\left(\epsilon\right)d\epsilon - \int_{\bar{g}_i^+(\tilde{\mathbf{x}})}^{g_i^-(\tilde{\mathbf{x}})} \epsilon\phi\left(\epsilon\right)d\epsilon\right).\end{aligned} \tag{B.4}$$

Note the following result holds:

$$\int_a^b x\phi(x)dx = \phi(a) - \phi(b). \tag{B.5}$$

Therefore, the equation (B.3) holds. ■

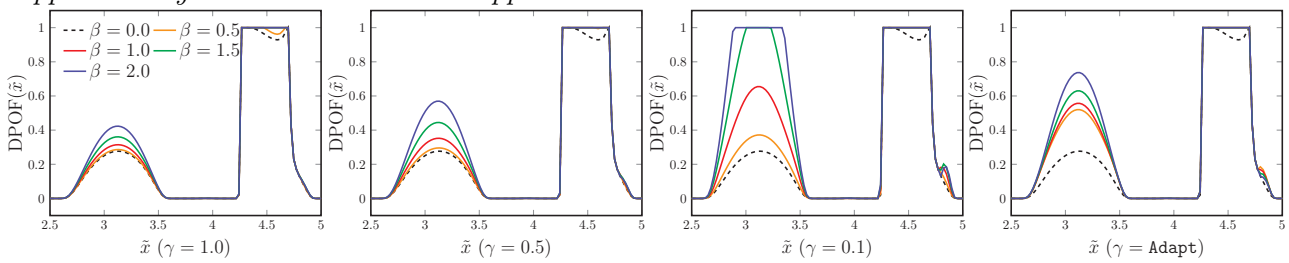

Figure B-1: Illustration of different exploration ability of DPOF with equation (B.3) and different settings of β and γ.

Based on Lemma 1, we propose a new exploration function as follows:

$$\rho^i(\tilde{\mathbf{x}}, \gamma^i) = \frac{\text{EMUB}^i(\tilde{\mathbf{x}})}{\gamma^i}, \tag{B.6}$$

where $\gamma^i > 0$ denotes a scale parameter.

As depicted in Fig. B-1, both γ^i and β can effect the exploration ability of equation (B.3). To mitigate the complexity of configuration, we introduce an adaptation law of γ^i as

$$\gamma^i = \texttt{Adapt}(\tilde{\mathbf{x}}) = \max_{\tilde{\mathbf{x}}} \text{EMUB}^i(\tilde{\mathbf{x}}) \Phi_g^i(\lambda). \tag{B.7}$$

This law enables γ to scale $\text{EMUB}^i(\tilde{\mathbf{x}})$ within $[0, 1]$. As shown in Fig. B-1, `Adapt` leads to a more reasonable strategy of weight allocation.

Remark 1. *The dissections to each term in equation (B.3) were made in [27] where the utility function was slightly different from equation (B.2) (they used a squared form as $\varepsilon^2(\tilde{\mathbf{x}}) - \tilde{g}_i^2(\tilde{\mathbf{x}})$ However, the derivation made by Ranjan et al. [27] was incorrect [28]). In general, the first and third terms in equation (B.3) suggest candidates on the most uncertainty boundaries, while the second term in equation (B.3) suggests candidates from the interior of regions that remain uncertain due to limited observations.*

Remark 2. *The alternative exploration function performs more aggressively because it explicitly contains the standard deviation σ_g^i that facilitates a reduction of global uncertainty of feasible regions. Unlike the POB and equation (10), EMUB is no longer a conservative design and may assign a high weight to a candidate solution that has low POF.*

B.3 Comparative study of EI-based CBO methods

Since this work does not aim at calibrating the involved hyperparameters, such as γ^i and β, for better performance in individual problems, we compare three EI-based CBO methods with fixed parameter settings: 1) `EIC` without hyperparameters, *ii*) `EICB-POB` using equation (10) with $\beta = 1.96$ (i.e. the 95% confidence level), and *iii*) `EICB-MUB` using equation (B.6) with $\beta = 1.96$ and γ^i determined by `Adapt`. Note that for `EICB-EMUB`, β and γ^i are coupled. Despite with `Adapt`, different values of β can still result in distinct optimization trajectories. This is one reason for our recommendation for equation (10) as the exploration function, without paying much effort on parameter selections.

We conduct four experiments in S1: 5D KBF, 10D KBF, XGB-H, and Lunar, as presented in Section 5.1 of the benchmark suite. The results are shown in Fig. B-2. It is observed that, for KBF, `EICB-MUB` does not outperform `EIC`. Differently, for XGB-H and Lunar, `EICB-MUB` is competitive with `EICB-POB`. Furthermore, it is noticed that the value deviation of `EICB-MUB` is larger than `EICB-POB` during optimization processes, which suggests that more exploration is introduced in `EICB-MUB`. However, this aggressive design can lead to inefficiency for some problems such as KBF and XGB-H, which is another reason for our preference of using equation (10). Note that we can expect improvements of `EICB-MUB` by appropriately tuning β and γ^i, which is however out of the scope of this work.

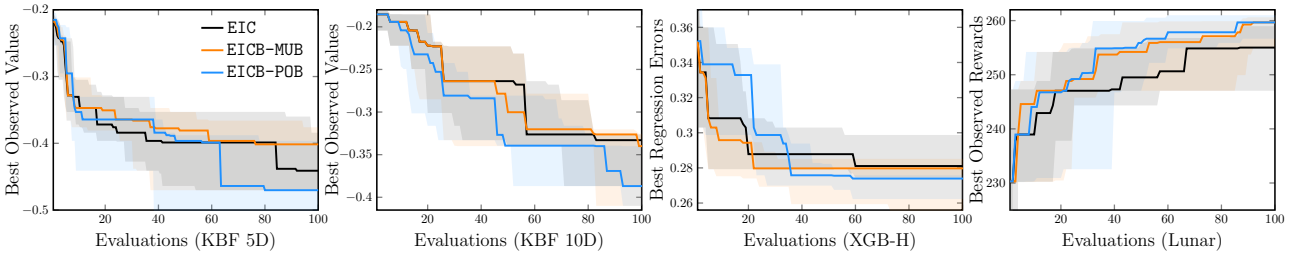

Figure B-2: Optimization trajectories of different tasks using EI-based methods in S1.

Section C. Expectation Propagation for HLGP

The posterior of the HLGP can be calculated by:

$$p(\tilde{\mathbf{g}}_i|X,\mathbf{g}_i) = \frac{1}{Z}p(\tilde{\mathbf{g}}_i|X)\prod_{k=1}^{N} p(g_i^k|\tilde{g}_i^k), \tag{C.1}$$

where Z is the normalization factor. For the k-th observation g_i^k, EP assigns it an un-normalized Gaussian distribution $t_i^k \triangleq \tilde{Z}_i^k \mathcal{N}\left(\tilde{\mu}_i^k, \tilde{\sigma}_i^{k\,2}\right)$ to locally approximate its exact likelihood. In this vein, the posterior is approximated by:

$$\begin{aligned} p(\tilde{\mathbf{g}}_i|X,\mathbf{g}_i) &\approx \frac{1}{Z_{\text{EP}}}p(\tilde{\mathbf{g}}_i|X)\prod_{k=1}^{N} t_i^k = \mathcal{N}(\boldsymbol{\mu}_g^i, \Sigma_g^i) \\ \text{with}\quad \boldsymbol{\mu}_g^i &= \Sigma_g^i \tilde{\Sigma}_g^{i\,-1}\tilde{\boldsymbol{\mu}}_g^i \quad\text{and}\quad \Sigma_g^i = (K + \tilde{\Sigma}_g^{i\,-1})^{-1}, \end{aligned} \tag{C.2}$$

where $\tilde{\boldsymbol{\mu}}_g^i = \left(\tilde{\mu}_i^1, \cdots, \tilde{\mu}_i^N\right)^\top$, $\tilde{\Sigma}_g^i$ denotes a diagonal matrix with the k-th element $\tilde{\sigma}_i^{k\,2}$, and Z_{EP} is the marginal likelihood. For HLGP, the site parameters $\tilde{Z}_i^k$, $\tilde{\mu}_i^k$ and $\tilde{\sigma}_i^k$ in t_i^k are determined by the following laws.

- *Law 1.* The site parameters of an exact Gaussian likelihood $\mathcal{N}(g_i(\mathbf{x}^k), \sigma^2)$ are directly assigned by

$$\tilde{Z}_i^k = 1, \quad \tilde{\mu}_i^k = g_i(\mathbf{x}^k), \quad \tilde{\sigma}_i^k = \sigma. \tag{C.3}$$

- *Law 2.* The site parameters of a non-Gaussian likelihood are computed by the moment match [33]. First, the marginal for $\tilde{g}_i^k$ from (C.2) is $\mathcal{N}(\bar{\mu}_k^i, \bar{\sigma}_k^{i\,2})$, where $\bar{\mu}_k^i$ and $\bar{\sigma}_k^{i\,2}$ denote the i-th element of $\boldsymbol{\mu}_g^i$ and i-th diagonal element of Σ_g^i. Then, the cavity parameters, $\bar{\mu}_{-k}^i$ and $\bar{\sigma}_{-k}^{i\,2}$, can be computed by

$$\begin{aligned} \bar{\mu}_{-k}^i &= \bar{\sigma}_{-k}^{i\,2}\left(\bar{\sigma}_k^{i\,-2}\bar{\mu}_k^i - \tilde{\mu}_i^k\tilde{\sigma}_i^{k\,-2}\right), \\ \bar{\sigma}_{-k}^{i\,2} &= \left(\bar{\sigma}_k^{i\,-2} - \tilde{\sigma}_i^{k\,-2}\right)^{-1}. \end{aligned} \tag{C.4}$$

The desired moments on the true likelihood are:

$$\begin{aligned} \hat{\mu}_i^k &= \bar{\mu}_{-k}^i + \frac{\bar{\sigma}_{-k}^{i\,2}\phi(z_i^k)}{\Phi(z_i^k)\sqrt{\alpha^2 + \bar{\sigma}_{-k}^{i\,2}}}, \\ \hat{\sigma}_i^{k\,2} &= \sigma_{-i}^{k\,2} - \frac{\bar{\sigma}_{-k}^{i\,4}\phi(z_i^k)}{\left(\alpha^2 + \bar{\sigma}_{-k}^{i\,2}\right)\Phi(z_i^k)}\left(z_i^k + \frac{\phi(z_i^k)}{\Phi(z_i^k)}\right), \\ \hat{Z}_i^k &= \Phi(z_i^k), \end{aligned} \tag{C.5}$$

where $z_i^k = \frac{\bar{\mu}_{-k}^i}{\sqrt{\alpha^2+\bar{\sigma}_{-k}^{i\,2}}}$. Henceforth, the site parameters can be computed by matching the above moments. Mathematically, it means

$$\begin{aligned} \tilde{\mu}_i^k =& \tilde{\sigma}_i^{k\,2}\left(\hat{\sigma}_i^{k\,-2}\hat{\mu}_i^k - \sigma_{-i}^{k\,-2}\mu_{-i}^k\right), \\ \tilde{\sigma}_i^{k\,2} =& \left(\hat{\sigma}_i^{k\,2} - \sigma_{-i}^{k\,-2}\right)^{-1}. \end{aligned} \tag{C.6}$$

As the un-normalized term $\tilde{Z}_i^k$ does not affect the modeling result, we omit its computation here. Detailed derivation of above processes is referred to [30, 33].

- *Law 3.* Repeatedly compute the above two laws for $k = 1, \ldots, N$, until all values of the site parameters converge.

Remark 3. *Computational instability and invalid operation may occur when updating the above parameters. We suggest several ways to alleviate this phenomenon. The first way is using calculation tricks with stronger numerical stability, refer to [30] for practical implementation of an EP algorithm. When non-singularity exists, one can fix the covariance matrices by manipulating the eigenvalues, see the implementation of [1]. Moreover, alternative choices of the Non-Gaussian likelihoods, e.g., step function studied in [3], can be more effective for a description of the truncated distribution.*

Section D. Experiment Settings

D.1 Algorithm Implementation

CBO methods

In our project, we implement all the algorithms on two widely used platforms, namely Gpflow [2] and GPy [3]. The `EIC` algorithm has been officially presented in Trieste [44]. The implementations of `TSC`[4] [19] and `MESC`[5] [13] are planted from their official open-source projects. For `MESC`, we choose to use 20 samples with a 1 000 grid in Thompson sampling. For `TSC`, we fix the grid number as 1 000. In our experiments, when increasing the number of samples and grids, we did not observe a significant improvement in the optimization efficiency. For `CBOB`, we fix $\beta = 1.96$ for 95% confidence level. Note that tuning β will not lead to significantly different behaviors of `CBOB` with the conservative exploration function designed in equation (10).

Gaussian processes

Uniformly, the GPR models are used to build the surrogate models of objective functions, where observations are sequentially obtained with a mask on infeasible solutions. We use GPR, GPC, and HLGP models to build models of constraints for specific algorithms and tasks. All GP models take a constant mean function and the Matérn 5/2 kernel. The GPC models are created by the default builder of Gpflow/GPy with initial observations [44] and optimized by the variational inference or EP. For HLGPs, we first modify the raw observations using the EP algorithm in consideration of different likelihood functions. Then the GP model with heterogeneous noises is utilized.

More configurations

All acquisition functions are optimized by the L-BFGS-B method with 1 000 iterations. The hyperparameters in all GP models are optimized according to the batch optimizer embedded in GPflow and GPy. All illustrative examples and experiments are performed on a desktop with Intel(R) Xeon(R) CPU E5-2620 v4 (2.10GHz) and NVIDIA GeForce GTX 1080Ti GPU.

[2] https://gpflow.github.io/GPflow
[3] https://gpy.readthedocs.io
[4] https://github.com/pytorch/botorch
[5] https://github.com/takeuchi-lab/CMES-IBO

Problem	Dimension	Objective	constraint	search space
KBF [36]	$n = 10,\ m = 2$	$-\lvert\frac{\sum_{i=1}^{10}\cos^4(x_i)-2\Pi_{i=1}^{10}\cos^2(x_i)}{\sqrt{\sum_{i=1}^{10} ix_i^2}}\rvert$	$0.75 - \Pi_{i=1}^{10} x_i < 0$ $\sum_{i=1}^{10} x_i - 75 < 0$	$[0, 10]^{10}$
Ackley [19]	$n = 10,\ m = 1$	$a = 20,\ b = 0.2,\ c = 2\pi$	$\sum_{i=1}^{10} x_i < 0$	$[-5, 5]^{10}$
WBD [37]	$n = 4,\ m = 5$	please refer to [37]		$x_1, x_4 \in [0.125, 5]$ $x_2, x_3 \in [0.1, 10]$
PVD [39]	$n = 4,\ m = 4$	please refer to [39]		$x_1, x_2 \in [0, 20]$ $x_3 \in [10, 50]$ $x_4 \in [150, 200]$

Table 3: Configurations of synthetic benchmark problems.

D.2 Synthetic Benchmark

This section delineates the configurations of individual synthetic benchmark problems. For brevity, we present a comprehensive table (Table 3) including the basic settings of each problem regarding black-box optimization.

KBF

The Keane bump function (KBF) is a synthetic constrained optimization problem for testing constrained optimization methods [36]. It consists of one objective function and two constraint functions. We consider the 10D KBF with the search space as $[0, 10]^{10}$.

Ackley

This problem is studied in [19]. The Ackley function is used with the recommended variable values[6]. The single constraint considered in this problem is $\sum_{i=1}^{10} x_i \leq 0$. Besides, the search space is given by $[-5.0, 5.0]^{10}$, and the optimum is located at the origin of the axes such that $f(\vec{0}) = 0$.

WBD

The standard definition of the welded beam design problem is given in [37]. We use the search space $x_1, x_4 \in [0.125, 5]$ and $x_2, x_3 \in [0.1, 10]$, which is also used in [39]. `EIC` finds a good solution within 20 FEs, but will be stagnated in the following evaluations. Differently, `CBOB` has less efficiency in the early stage, but can gradually find a good solution without prolonged stagnation.

PVD

The standard definition of the pressure vessel design problem can be found in [39]. The search space is adopted from [19] as $x_1 \in [0, 20]$, $x_2 \in [0, 20]$, $x_3 \in [10, 50]$ and $x_4 \in [150, 200]$. To deal with the discrete variables x_1 and x_2, we uniformly round them to be integers. In this experiment, $\texttt{MESC}_c$ fails to find a better solution within 100 function evaluations, therefore omitted from the final results.

D.3 Real-World Benchmark

We consider two kinds of real-world problems, i.e., the hyperparameter optimization (HPO) for a good machine learning model and the reinforcement learning for a better controller. Therein, we consider the model size and energy consumption as the analytically unknown constraints. The constraint thresholds are determined by the mean values of the model size or total energy after 2 000 repetitive experiments with randomly sampled solutions. Therefore, about 50% solutions of the initial evaluations are infeasible. A brief table (Table 4) is given with basic configurations of each problem.

[6] https://www.sfu.ca/~ssurjano/ackley.html

Problem	Dimension	Objective	constraint	threshold	search space
XGB-H	$n = 7$ $m = 1$	MSE	model size	50 000	learning rate: $[2^{-10}, 1]$ (log) maximum depth: $[1, 15]$ (Int) subsample ratio: $[0.01, 1]$ L2 regularization: $[2^{-10}, 2^{10}]$ (log) L1 regularization: $[2^{-10}, 2^{10}]$ (log) minimum weight sum: $[1, 2^7]$ (log) estimator number: $[1, 2^8]$ (log, Int)
MLP-D	$n = 8$ $m = 1$	fraction of correctly classified samples	model size	107 000	learning rate: $[10^{-5}, 1]$ (log) hidden layer 1: $[2^2, 2^8]$ (log, Int) hidden layer 2: $[2^2, 2^8]$ (log, Int) batch size: $[2^2, 2^8]$ (log, Int) L2 regularization: $[10^{-8}, 10^{-3}]$ (log) Adam decay rate 1: $[0, 0.9999]$ (log) Adam decay rate 2: $[0, 0.9999]$ (log) tolerance: $[10^{-6}, 10^{-2}]$ (log)
Lunar	$n = 12$ $m = 1$	reward	fuel cost	40.0	tolerance: $[0, 2]^{12}$
Swim.	$n = 16$ $m = 1$	reward	energy cost	1.2	$[-1, 1]^16$

Table 4: Configurations of synthetic benchmark problems.

XGB-H

The California housing is a regression problem with 20 640 samples. The dataset is prepared with a 0.25 train/test split in scikit-learn. The XGBoost regressor[7] is used to fit the dataset, for which we optimize 7 configurable parameters with different algorithms. The parameters include: learning rate $[2^{-10}, 1]$ (log), maximum depth $[1, 15]$ (Int), subsample ratio of columns $[0.01, 1]$, L2 regularization term $[2^{-10}, 2^{10}]$ (log), L1 regularization term $[2^{-10}, 2^{10}]$ (log), minimum sum of instance weight in a child $[1, 2^7]$ (log), and number of estimators $[1, 2^8]$ (log, Int). We round the solutions for integer parameters. With 2 000 repetitive experiments, the model size threshold is 50 000bytes.

MLP-D

The digits dataset contains 1 797 images of handwriting digits from 1-10. It is a common classification problem and prepared with a 0.25 train/test split in scikit-learn. We use a Multi-layer Perceptron (MLP) classifier to fit the dataset, for which we optimize 8 configurable parameters with different algorithms. The parameters include: initial learning rate $[10^{-5}, 1]$ (log), two sizes of the hidden layers $[2^2, 2^8]^2$ (log, Int), batch size $[2^2, 2^8]$ (log, Int), L2 regularization term $[10^{-8}, 10^{-3}]$ (log), two exponential decay rates in Adam $[0, 0.9999]^2$, tolerance for the optimization $[10^{-6}, 10^{-2}]$ (log),. With 2 000 repetitive experiments, the model size threshold is 107 000bytes.

Lunar

In this experiment, we should design a controller that renders a rocket to land on the target position with more reward determined by *i*) the distance to the target, *ii*) fuel consumption, and *iii*) whether crashed. As studied in [38], a heuristic controller with 12 parameters can be designed to control the rocket. We use 'LunarLander-v2' in simulations, where the fuel consumption is separated as an energy constraint. In this task, the fuel consumption should be less than 40.0. The search space is $[0, 2]^{12}$. To create a deterministic environment, we fix the terrain and initial state of the rocket.

[7] https://github.com/dmlc/xgboost/

Swimmer

In this experiment, the goal is to control a swimmer to move as fast as possible towards one direction. In the MujoCo environment [45], we use 'swimmer-v4' which contains 8D output state space and 2D action space. As designed in [40], the linear feedback controller is utilized, resulting in 16 parameters in the control gain matrix to be determined. Similarly, we separate the control cost, which was considered in the total reward, as an energy constraint. The constraint threshold is set as 1.2. The search space is set as $[-1, 1]^{16}$. To obtain a robust controller against small noises, in each evaluation, the simulation runs 5 times with different initialization seeds.

E. Further Discussions

E.1 Limitations

Our investigations of EICB have two limitations. First, we only study the asymptotic convergence of EICB, from which we cannot give definite answers on whether EICB outperforms EIC at the convergence rate in theory. On the other hand, the empirical design of exploration functions needs to be further studied for better guidance to the practitioners.

In addition to EICB, the design of HLGP models can be further improved in consideration of the uncertainty on observed solutions. Specifically, the HLGP in its current form cannot eliminate the uncertainty of an evaluated infeasible point, as depicted in Fig. 2. Note that GPC also fails to eliminate the uncertainty. Differently, a classifier such as SVM or MLP with appropriate model capacity can easily achieve this goal. We believe a more complex design of the surrogate models can be leveraged to better exploit the mixed observations.

E.2 Potential Impact

The proposed DPOF is a new constraint-handling technique that can be easily integrated with non-negative acquisition functions, such as EI, probability of improvement, parzen estimator, and density-ratio estimation. With other acquisition functions, the theoretical analysis on both convergence and regret bound may be conducted, potentially leading to a more principled design of DPOF. Besides, our work gives criteria for the research focusing on the adaptive design of CBO methods by leveraging information from similar tasks for better optimization with unknown constraints.