A General Recipe for Likelihood-free Bayesian Optimization

Jiaming Song [* 1]   Lantao Yu [* 2]   Willie Neiswanger [2]   Stefano Ermon [2]

Abstract

The acquisition function, a critical component in Bayesian optimization (BO), can often be written as the expectation of a utility function under a surrogate model. However, to ensure that acquisition functions are tractable to optimize, restrictions must be placed on the surrogate model and utility function. To extend BO to a broader class of models and utilities, we propose likelihood-free BO (LFBO), an approach based on likelihood-free inference. LFBO directly models the acquisition function without having to separately perform inference with a probabilistic surrogate model. We show that computing the acquisition function in LFBO can be reduced to optimizing a weighted classification problem, where the weights correspond to the utility being chosen. By choosing the utility function for expected improvement (EI), LFBO outperforms various state-of-the-art black-box optimization methods on several real-world optimization problems. LFBO can also effectively leverage composite structures of the objective function, which further improves its regret by several orders of magnitude.

1. Introduction

Bayesian optimization (BO) is a framework for global optimization of black-box functions, whose evaluations are expensive. Originated from applications on engineering design (Kushner, 1962; 1964; Močkus, 1975), BO has seen successes in various domains, including automated machine learning (Bergstra et al., 2011; Snoek et al., 2012; Swersky et al., 2013), simulation optimization (Pearce & Branke, 2017; Wang et al., 2020), drug discovery (Griffiths & Hernández-Lobato, 2020), graphics (Brochu et al., 2007;

Koyama et al., 2017), robotics (Calandra et al., 2016), and battery charging protocols design (Attia et al., 2020).

A cornerstone of BO methods is the acquisition function that determines which candidates to propose for function evaluations. Many acquisition functions can be written in the form of the expectation of a utility function over a surrogate model (Wilson et al., 2018; Garnett, 2022). To evaluate and optimize these acquisition functions effectively, the surrogate model is often chosen to produce tractable probability estimates (*e.g.*, Gaussian processes (Rasmussen, 2003)), and the utility functions are often chosen such that the expectations can be computed analytically, such as Probability of Improvement (PI, (Kushner, 1964)).

However, the requirements of tractability and efficiency often exclude popular families of surrogate models and utility functions relevant to the optimization problem. For instance, implicit models defined via simulators (Diggle & Gratton, 1984) may fit the data-generating process better but do not have tractable probability estimates. Meanwhile, objectives that are defined over many correlated outcomes are often partially known (*i.e.*, in a grey-box BO setting (Astudillo & Frazier, 2022; Maddox et al., 2021)), yet utilities defined for these objective functions may not have tractable integrals even for Gaussian process surrogate models. While Monte Carlo methods can be employed here (Astudillo & Frazier, 2019; Kleinegesse & Gutmann, 2019), they scale poorly when many samples are needed for low variance estimates.

Inspired by how likelihood-free density ratio estimation extends variational inference to implicit models (Sugiyama et al., 2012; Tran et al., 2017), we describe a general approach to obtain "likelihood-free" acquisition functions that extend BO to a broader class of models and utilities. We develop a likelihood-free method that directly estimates the ratio between any pair of non-negative measures; this includes any acquisition function with a non-negative utility function, and allows us to directly model it without having to separately perform inference with a probabilistic surrogate model. Our likelihood-free BO (LFBO) approach can be reduced to a weighted classification problem, which is easy to implement for general utility functions.

We evaluate our LFBO method empirically on a number of synthetic and real optimization problems. On several real-world benchmarks, LFBO outperforms various state-of-
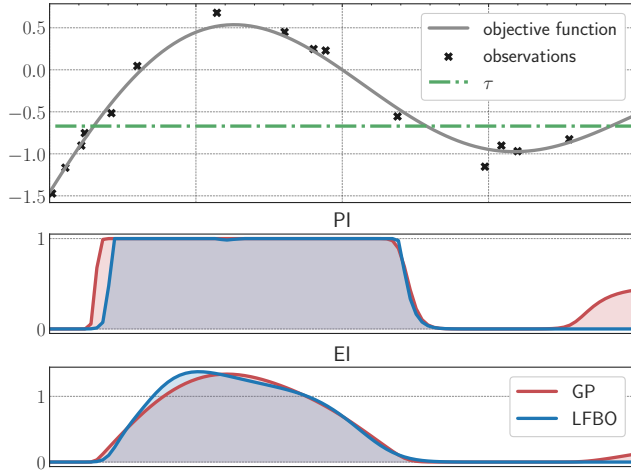
---

*Equal contribution   [1]NVIDIA (Work done while at Stanford). [2]Stanford University. Correspondence to: Jiaming Song <jiamings@nvidia.com>.

Figure 1. BO over a synthetic function $g(x) = -\sin(3x) - x^2 + 0.6x$ with noise $\mathcal{N}(0, 0.01)$. (*Top*) the objective function $g(x)$ with $n = 15$ noisy observations and the threshold τ. (*Bottom*) the evaluated acquisition function values with Gaussian Process (GP, *red*) and our likelihood-free BO (LFBO, *blue*) over PI and EI utility functions. Our LFBO acquisition function does not separately perform probabilistic inference over a surrogate model and can adapt to any non-negative utility function.

the-art methods in black-box optimization. On two neural network tuning tasks, we show that LFBO can further be improved with other novel utility functions. On an environmental modeling problem (Astudillo & Frazier, 2019), LFBO can leverage the special composite structure of the objective function and improve its optimization efficiency.

2. Background

Bayesian optimization (BO) aims to find a strategy that effectively maximizes a black-box function $g : \mathcal{X} \to \mathbb{R}$, given available information regarding g as a set of n observations $\mathcal{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$; it is commonly assumed that $y_i = g(\mathbf{x}_i) + \epsilon$ is observed with noise ϵ. In sequential decision making settings, at each iteration i, BO methods select the next query $\mathbf{x}_i$ via maximizing an acquisition function $\mathcal{L} : \mathcal{X} \to \mathbb{R}$, which utilizes a surrogate model $\mathcal{M}$.

$\mathcal{M}$ may provide a probabilistic interpretation of g, which can then be used to evaluate the acquisition function at any point $\mathbf{x} \in \mathcal{X}$. After updating the model parameters based on $\mathcal{D}_n$, a belief over $g(\mathbf{x})$ can be formed as $p(y|\mathbf{x}, \mathcal{D}_n)$. The Gaussian process (GP, Rasmussen (2003)) is a popular surrogate model due to its posterior being analytically tractable; however, the basic form of GP requires $\mathcal{O}(n^3)$ computational complexity to perform posterior inference, making it difficult to scale with large observation sets. Although researchers have proposed sparse GPs with approximations for improved scalability, the inference time still scales with

the number of data points (e.g., $\mathcal{O}(n)$ in (Titsias, 2009)).

Acquisition functions based on expected utility. Many acquisition functions at an input location $\mathbf{x}$ are defined as an expected utility (EU) over the posterior belief $p(y|\mathbf{x}, \mathcal{D}_n)$:

$$\mathcal{L}^{(u)}(\mathbf{x}; \mathcal{D}_n, \tau) = \mathbb{E}_{y \sim p(y|\mathbf{x}, \mathcal{D}_n)}[u(y; \tau)] \tag{1}$$

$$= \int u(y; \tau) p(y|\mathbf{x}, \mathcal{D}_n) \mathrm{d}y, \tag{2}$$

where $u(y; \tau)$ is a chosen utility function with hyperparameter $\tau \in \mathcal{T}$ ($\mathcal{T}$ is the set of allowed values) that specifies the utility of observing y at $\mathbf{x}$ and controls the exploration-exploitation trade-off (Wilson et al., 2018; Garnett, 2022). The most common examples are Probability of Improvement (PI, Kushner (1964)) whose utility indicates whether y exceeds τ:

$$u^{\mathrm{PI}}(y; \tau) := \mathbb{I}(y - \tau > 0), \tag{3}$$

where $\mathbb{I}$ is the binary indicator function; and Expected Improvement (EI, Mockus et al. (1978)) whose utility indicates how much y exceeds the threshold τ:

$$u^{\mathrm{EI}}(y; \tau) := \max(y - \tau, 0). \tag{4}$$

Other examples include Entropy Search (ES, Hennig & Schuler (2012); Hernández-Lobato et al. (2014)) and Knowledge Gradient (Frazier et al., 2008; Scott et al., 2011), whose utilities are more complicated expectations themselves.

Acquisition functions based on density ratios. While most methods used in BO model the distribution of outcomes of the black-box function g, some methods do not use an explicit model in the form of $p(y|\mathbf{x}, \mathcal{D}_n)$, and instead model the acquisition function directly via density ratios over $\mathbf{x}$. Density ratio (DR) acquisition functions (Bergstra et al., 2011) expresses the acquisition function with the ratio of two model densities[1] over $\mathbf{x}$, $p(\mathbf{x}|y > \tau, \mathcal{D}_n)$ and $p(\mathbf{x}|y \le \tau, \mathcal{D}_n)$, that models the densities of $\mathbf{x}$ conditioned on y being above or below the threshold τ, respectively[2]:

$$L^{\mathrm{DR}}(\mathbf{x}; \mathcal{D}_n, \tau) = \frac{p(\mathbf{x}|y > \tau, \mathcal{D}_n)}{p(\mathbf{x}|\mathcal{D}_n)} \tag{5}$$

$$= \frac{p(\mathbf{x}|y > \tau, \mathcal{D}_n)}{p(\mathbf{x}|y > \tau, \mathcal{D}_n)\gamma_{\mathcal{D}_n} + p(\mathbf{x}|y \le \tau, \mathcal{D}_n)(1 - \gamma_{\mathcal{D}_n})},$$

where $\gamma_{\mathcal{D}_n} := p(y > \tau|\mathcal{D}_n)$. Intuitively, when the ratio in Equation 5 is high for a certain $\mathbf{x}$, it is believed to produce

---

[1]We overload the term "density" to include "probability" for discrete measures; we also consider maximization instead of minimization of objectives.

[2]Throughout the paper, we use the notation $\mathcal{L}$ to indicate acquisition functions that explicitly use tractable forms of $p(y|\mathbf{x}, \mathcal{D}_n)$, and the notation L to indicate acquisition functions that do not (*e.g.*, acquisition functions based on density ratios).

favorable outcomes (larger than τ) with a high probability; this makes density ratios being successfully used in hyperparameter tuning and neural architecture search (Bergstra et al., 2013).

To estimate this density ratio, Bergstra et al. estimate the numerator and denominator with two seprate Tree Parzen Estimators (TPE), whereas Bayesian optimization by density-ratio estimation (BORE, Tiao et al. (2021)) estimate this density ratio directly with likelihood-free inference (Sugiyama et al., 2012). Bergstra et al. and Tiao et al. further argue that the DR acquisition function is equivalent to the EI acquisition function up to some constant factor; thus DR is limited to one particular type of utility function (which is supposedly EI). In Section 3, however, we prove that DR is equivalent to PI instead of EI, and then propose a solution to generalize DR-based BO to other acquisition functions, including EI.

3. Likelihood-free Bayesian Optimization

As discussed earlier, the acquisition function is constructed by two components, the surrogate model and the utility function[3], over which restrictions are often placed to ensure that the acquisition function is tractable and efficient to compute. To allow tractable inference, EU acquisition functions often choose models such that $p(y|\mathbf{x}, \mathcal{D})$ is explicitly defined (*e.g.*, Gaussian), which excludes other popular models such as implicit, simulator-based models (Diggle & Gratton, 1984) that do not have tractable likelihood estimates. While DR acquisition functions do not pose such limitations, they are limited in terms of the utility function and cannot effectively leverage knowledge of the structure of function $g(\mathbf{x})$ in a grey-box BO setting (Astudillo & Frazier, 2022).

Inspired by how likelihood-free variational inference (VI) extends VI to implicit models, we develop "likelihood-free" acquisition functions that extend BO to more general models and utility functions. First, we introduce a formal definition for when a likelihood-free acquisition function admits a certain utility function (Section 3.1). We then develop likelihood-free acquisition functions that can admit any non-negative utility function (Section 3.2) and discuss a special case that amounts to weighted classification (Section 3.3). Finally, we discuss the representation capability of likelihood-free acquisition functions (Section 3.4).

3.1. Equivalence between Acquisition Functions

Before we establish our likelihood-free BO method, we introduce an equivalence relationship among acquisition functions that describes if two acquisition functions would behave identically under infinite observations, even with different types of surrogate models. This allows us to replace

the expected utility acquisition function with an equivalent likelihood-free approach in practical scenarios, where the difference in exploration-exploitation trade-offs would only occur from the difference in models fitted from finite queries.

Definition 3.1 (Equivalence between acquisition functions). Assume that the statistical models that we consider are consistent (satisfying Assumption A.3 in Appendix A). For all $n \in \mathbb{N}$, let $\mathcal{D}_n^{(1)}$ and $\mathcal{D}_n^{(2)}$ denote the random variables representing n non-*i.i.d.* queries from two valid sequential decision making processes for the same black-box function (satisfying Definition A.1 in Appendix A). Let $\mathrm{Ac}^{(1)}$ and $\mathrm{Ac}^{(2)}$ denote two acquisition functions with the same search space $\mathcal{X}$ and hyperparameter space $\mathcal{T}$. We say that "$\mathrm{Ac}^{(1)}$ is equivalent to $\mathrm{Ac}^{(2)}$" if $\forall \mathbf{x} \in \mathcal{X}, \tau \in \mathcal{T}$, a positive constant factor α, and a constant β, the following is true:

$$\operatorname*{plim}_{n \to \infty} \mathrm{Ac}^{(1)}(\mathbf{x}; \mathcal{D}_n^{(1)}, \tau) = \alpha \operatorname*{plim}_{n \to \infty} \mathrm{Ac}^{(2)}(\mathbf{x}; \mathcal{D}_n^{(2)}, \tau) + \beta.$$

We note that for a sequential decision process to be valid (Definition A.1), it needs to place non-zero probabilities to every point in the search space. Since any ϵ-greedy strategy with $\epsilon > 0$ would satisfy this, our condition here is reasonable (as we can apply ϵ-greedy strategy to any decision process based on acquisition functions). Moreover, a similar consistency assumption is often made for GP-based methods (Williams & Rasmussen, 2006).

With this definition, the claim that DR acquisition functions being equivalent to the EI acquisition function (Bergstra et al., 2011; Tiao et al., 2021) may be interpreted as "L^{DR} is equivalent to $\mathcal{L}^{\mathrm{EI}}$". Unfortunately, this interpretation is inaccurate according to our definitions. In Proposition 3.2, we introduce an equivalence result between DR and PI. This echos the claim made in Garnett (2022).

Proposition 3.2. L^{DR} *is equivalent to* $\mathcal{L}^{\mathrm{PI}}$.

Proof. In Appendix A.2. □

The above statement indicates that the density ratio acquisition functions are equivalent to probability of improvement under the same τ, which justifies its practical effectiveness. However, as PI and EI are clearly different acquisition functions, DR cannot be equivalent to EI[4]. We include a formal statement in Corollary A.4 and its proof in Appendix A.

Another intuitive argument for the non-equivalence between DR and EI is through the weights of the observed y in the acquisition function: for a fixed threshold τ, DR would treat all $(\mathbf{x}, y)$ pairs above it with equal importance (just like how PI does), whereas EI also considers how much y is greater than τ and places more importance when $(y - \tau)$ is higher.

---

[3]In the case of DR, the utility function is implicitly chosen.

[4]We note that in Tiao et al. (2021), the "equivalence" between DR and EI are established only for specific surrogate models.
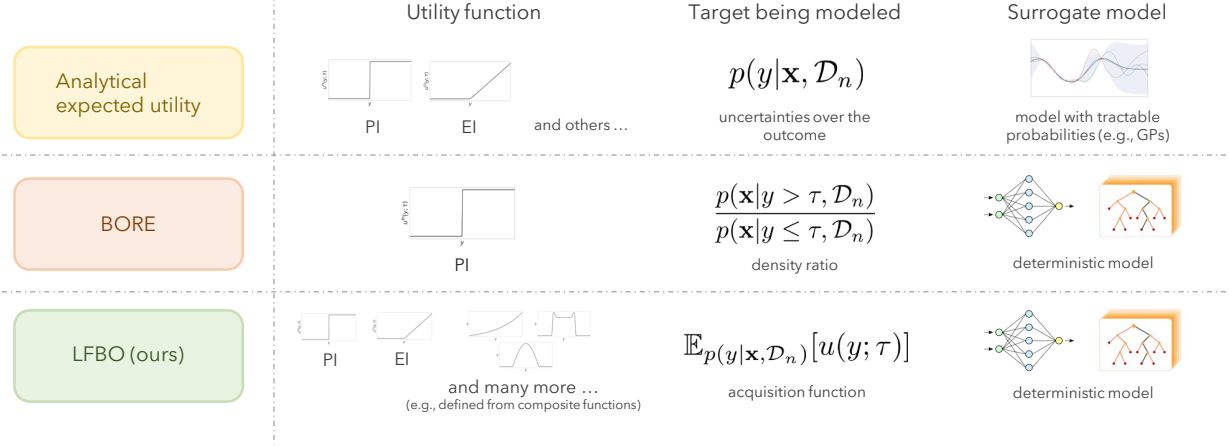
Figure 2. Overview of several types of acquisition functions for BO. (*Top*) Closed-form acquisition functions in the form of expected utility often require certain utility functions (*e.g.*, PI and EI) and certain surrogate models (*e.g.*, GPs). (*Middle*) BORE (Tiao et al., 2021) uses simple, deterministic models (*e.g.*, neural networks, decision trees) to model a density ratio with likelihood-free inference, but this limits its utility function to the one associated with PI. (*Bottom*) Our likelihood-free BO approach extends BORE to any non-negative utility function, including the one associated with EI and others whose expectations may not have a closed-form solution (*e.g.*, composite function EI in (Astudillo & Frazier, 2019)).

3.2. A General Recipe for Likelihood-free BO

Given the above results, it is natural to ask if there exists another likelihood-free acquisition function that is actually equivalent to EI. Here, we provide a general approach to obtain likelihood-free acquisition functions that are equivalent to acquisition functions based on any non-negative expected utility function, which naturally includes both EI and PI.

Our approach is inspired by density ratio estimation via variational f-divergence estimation (Nguyen et al., 2008) but generalizes it beyond the ratio between densities. The following lemma provides a variational representation for the expected utility at any point $\mathbf{x}$, provided with samples from some $p(y|\mathbf{x})$ (that may or may not depend on data). This enables us to replace integration (that are potentially intractable for complex $p(y|\mathbf{x})$ and u) with a variational objective (that can be performed on samples).

Lemma 3.3 (Variational representation of an integral). *For all non-negative utility functions $u : \mathbb{R} \times \mathcal{T} \to [0, \infty)$, and for all strictly convex functions $f : [0, \infty) \to \mathbb{R}$ with third order derivatives,*

$$\mathbb{E}_{p(y|\mathbf{x})}[u(y; \tau)] =$$
$$\arg\max_{s \in [0, \infty)} \mathbb{E}_{p(y|\mathbf{x})}[u(y; \tau)f'(s) - f^\star(f'(s)),$$

where $f^\star$ is the convex conjugate of f, and the maximization is performed over the $s \in [0, \infty)$ (the domain of f).

Proof. In Appendix A.3. □

By minimizing a variational objective averaged under any

distribution that is supported on the search space $\mathcal{X}$, we can recover an acquisition function over $\mathbf{x}$ that is equivalent to any expected utility acquisition function in the form of Equation 2. This new acquisition function is likelihood-free, in the sense that it does not model distributions with a tractable probability and only uses samples from the observations $\mathcal{D}_n$.

Theorem 3.4. *For u and f as defined in Lemma 3.3, $\forall \mathbf{x} \in \mathcal{X}$ and $\forall \tau \in \mathcal{T}$, let $L_{\mathrm{LFBO}}^{(u)}(\mathbf{x}; \mathcal{D}_n, \tau) = \hat{S}_{\mathcal{D}_n, \tau}(\mathbf{x})$, where*

$$\hat{S}_{\mathcal{D}_n, \tau} = \arg\max_{S:\mathcal{X}\to\mathbb{R}} \mathbb{E}_{\mathcal{D}_n}[u(y; \tau)f'(S(\mathbf{x})) - f^\star(f'(S(\mathbf{x})))].$$

Then $L_{\mathrm{LFBO}}^{(u)}$ is equivalent to $\mathcal{L}^{(u)}$ in Equation 2.

Proof. In Appendix A.3. □

The above statement converts the process of obtaining the acquisition function into an optimization problem that only relies on samples, such that the optimized model $\hat{S}_{\mathcal{D}_n, \tau}$ can directly be used as the acquisition function. Therefore, we have obtained a general approach to construct acquisition functions that are (*i*) likelihood-free and (*ii*) equivalent to an expected utility acquisition function. Due to the likelihood-free nature of the method, we name our approach Likelihood-free Bayesian Optimization (LFBO).

3.3. Practical Likelihood-free Acquisition Functions

From Theorem 3.4, we can further choose specific convex f and recover practical objective functions such as a weighted classification one.

Corollary 3.5 (Weighted classification). *For u as defined in Lemma 3.3, $\forall \mathbf{x} \in \mathcal{X}$ and $\forall \tau \in \mathcal{T}$, let*

$$L_{\mathrm{C}}^{(u)}(\mathbf{x}; \mathcal{D}_n, \tau) = \hat{C}_{\mathcal{D}_n, \tau}(\mathbf{x}) / (1 - \hat{C}_{\mathcal{D}_n, \tau}(\mathbf{x})), \quad (6)$$

where $\hat{C}_{\mathcal{D}_n, \tau}$ is the maximizer to the following objective over $C : \mathcal{X} \to (0, 1)$:

$$\mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_n}[u(y; \tau) \log C(\mathbf{x}) + \log(1 - C(\mathbf{x}))], \quad (7)$$

then $L_{\mathrm{C}}^{(u)}$ is equivalent to $\mathcal{L}^{(u)}$ in Equation 2.

Proof. Let $f(r) = r \log \frac{r}{r+1} + \log \frac{1}{r+1}$ for all $r > 0$, then $f^\star(c) = -\log(1 - \exp(c))$, $f'(r) = \log r - \log(r+1)$, and $f''(r) = 1/r - 1/(r+1) > 0$ (so f is convex, and its third order derivative exists). Applying $f(r)$ and $S = C/(1-C)$ to Theorem 3.4 completes the proof. $\square$

From Corollary 3.5, we can optimize a classification objective (Equation 7) with weights defined by the utility function and expect it to behave similarly to the EU acquisition functions in Equation 2.

LFBO for PI   For PI, $u^{\mathrm{PI}}(y; \tau) := \mathbb{I}(y - \tau > 0)$, Equation 7 becomes:

$$\frac{|\mathcal{D}_n^+|}{|\mathcal{D}_n|} \underbrace{\mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_n^+}[\log C(\mathbf{x})]}_{\text{positive}} + \underbrace{\mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_n}[\log(1 - C(\mathbf{x}))]}_{\text{negative}},$$

where $\mathcal{D}_n^+$ denotes the subset of $\mathcal{D}_n$ where $y > \tau$. Similar to word2vec (Mikolov et al., 2013), we can treat samples in the first term as positive and samples in the second term as negative. Our objective encourages C to assign higher values to $\mathbf{x}$ in positive samples and lower values to $\mathbf{x}$ in negative samples; all the positive samples have the same weight, which is also the case in BORE.

LFBO for EI   For EI, $u^{\mathrm{EI}}(y; \tau) := \max(y - \tau, 0)$, Equation 7 becomes:

$$\frac{|\mathcal{D}_n^+|}{|\mathcal{D}_n|} \underbrace{\mathbb{E}_{\mathcal{D}_n^+}[(y - \tau) \log C(\mathbf{x})]}_{\text{positive}} + \underbrace{\mathbb{E}_{\mathcal{D}_n}[\log(1 - C(\mathbf{x}))]}_{\text{negative}},$$

where positive samples are weighted by $(y - \tau)$ times a constant ($|\mathcal{D}_n^+|/|\mathcal{D}_n|$). Different from PI, here we favor y that are not only above the threshold τ, but also have higher values. In practice, the factor $|\mathcal{D}_n^+|/|\mathcal{D}_n|$ can be removed, as this amounts to simply scaling the utility function by a constant (since the utilities/weights for the samples with $y < \tau$ are zero). We can also normalize the weights within the positive term such that the average weight is one; we adopt this approach in the experiments.

3.4. Representation Capability of Likelihood-free Acquisition Functions

Unlike GPs that produce a distributional estimate given $\mathbf{x}$, models used in likelihood-free acquisition functions (*e.g.*, $C(\mathbf{x})$ in Equation 7) would only produce a point estimate given $\mathbf{x}$. Despite the lack of explicit uncertainty estimates over y, likelihood-free acquisition functions are not limited in what they can represent; for example, wide-enough neural networks can approximate any acquisition function that is Borel measurable (Hornik et al., 1989). Conversely, even for the same utility u, there may exist many different surrogate models of $p(y|\mathbf{x}, \mathcal{D})$ that give the same acquisition value (see an argument for Gaussian distributions in Proposition A.6). Thus, our likelihood-free method is a more direct approach to acquiring the acquisition function without loss of representation capabilities.

4. Related Work

Bayesian Optimization   Many works in Bayesian Optimization adopt GPs as the underlying surrogate model and some form of expected utility as the acquisition function. The utilities in some acquisition functions can be expectations themselves (such as ES). Some works aim to make the posterior inference process more scalable, possibly sacrificing the tractability or the exactness of the posterior; these include sparse online GPs (McIntire et al., 2016), linear models (Perrone et al., 2018), neural network ensembles (White et al., 2019), Bayesian neural networks (Springenberg et al., 2016), Bayesian linear models and GPs with neural feature learners (Snoek et al., 2015; Tran et al., 2020), and GPs with deep kernels from meta learning (Wistuba & Grabocka, 2021). Nevertheless, these methods replace GPs with another model and do not change how they interact with the acquisition functions. TPE (Bergstra et al., 2011) is an exception in that it first estimates probability distributions over $\mathbf{x}$ (as opposed to over y given $\mathbf{x}$) and then obtains the acquisition function through density ratios over $\mathbf{x}$. Ratio estimation approaches have also been combined with mutual information acquisition functions for Bayesian experimental design (Kleinegesse & Gutmann, 2019; Zhang et al., 2021b).

The work most related to ours is BORE (Tiao et al., 2021), which applies likelihood-free density ratio estimation to the TPE acquisition function. Our approach extends BORE over two aspects: (*i*) we formally identify that BORE (and equivalently, TPE) is not equivalent to EI, and that it is only equivalent to the PI utility function; (*ii*) we derive a likelihood-free approach that applies to not only EI, but also general expected utility functions.

Likelihood-free inference   Likelihood-free inference methods have been widely applied to implicit models where

tractable forms of the likelihood do not exist, such as generative adversarial networks (GANs, Goodfellow et al. (2014)) and hierarchical implicit models (Tran et al., 2017). These are often posed as density/likelihood ratio estimation problems from samples. Various methods have been proposed beyond direct kernel density estimation approaches, such as KL importance estimation procedure (KLIEP, Sugiyama et al. (2007; 2012)), kernel mean matching (KMM, Gretton et al. (2009)), and unconstrained least-squared importance fitting (ULSIF, (Kanamori et al., 2010)). The likelihood-ratio estimation approach used in GANs can be interpreted as class probability estimation based on proper scoring rules (Buja et al., 2005; Gneiting & Raftery, 2007), variational f-divergence estimation (Nguyen et al., 2008), or ratio matching (Sugiyama et al., 2013; Uehara et al., 2016). These perspectives are known to be equivalent to one another (Reid et al., 2011; Mohamed & Lakshminarayanan, 2016) and are also applied in various domains such as generative modeling (Gutmann & Hyvarinen, 2010; Song & Ermon, 2020; Yu et al., 2020), domain adaptation (Bickel et al., 2007), inverse reinforcement learning (Yu et al., 2019), and black-box sequence design (Zhang et al., 2021a).

Our likelihood-free acquisition function is different from these prior approaches (such as class probability estimation used in BORE) in that we are not necessarily estimating the ratio between two probability measures defined over $\mathcal{X}$ (except for the case of PI), but the ratio between a general non-negative measure (defined as some expected utility over the positive samples) and a probability measure (over the negative samples). Nevertheless, our approach is closely related to these perspectives, and can be treated as a classification problem with unnormalized weights determined by the utility values.

5. Experiments

In this section, we evaluate our likelihood-free BO approach on multiple synthetic and real-world optimization tasks. We first verify our theory by comparing against ground truth targets in a synthetic case (Section 5.1). Next, we compare against various baselines, especially BORE (Tiao et al., 2021), illustrating the advantages of the EI utility function (as well as others) in conjunction with our likelihood-free BO approach (Section 5.2). Finally, we illustrate how we can massively improve performance by integrating LFBO with existing knowledge about the objective function, specifically with a composite, grey-box function optimization problem (Section 5.3). Here, LFBO uses the weighted classification approach in Equation 7, Corollary 3.5; we leave the possibilities of applying different f in Theorem 3.4 as future work. For all BO experiments, we first query 10 random candidates uniformly over the search space, and then proceed with the BO methods.
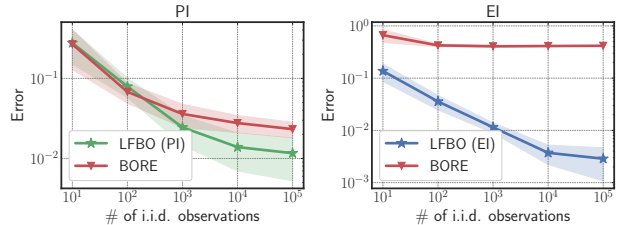

Figure 3. L_1 distance between the modeled acquisition functions and the ground truth. x is drawn uniformly between $[-1, 1]$. The shaded regions show the mean plus and minus one standard deviation across different seeds. The ground truth acquisition functions for the left and right figures are Probability of Improvement and Expected Improvement respectively.

5.1. Verifying the Equivalence Relationship of Likelihood-free Acquisition Functions

We first validate the theory over our proposed likelihood-free acquisition functions over a synthetic problem, where the search space is $x \in [-1, 1]$ and observations are obtained from $g(x) = -\sin(3x) - x^2 + 0.6x$ plus independent Gaussian noise $\epsilon \sim \mathcal{N}(0, 0.01)$ (see Figure 1 for the function). This allows to evaluate the "ground truth" EI and PI values for any x in the search space (similar to how EI and PI are computed for GPs). We estimate the ground truth EI and PI with our LFBO acquisition functions, with BORE (Tiao et al., 2021) as a baseline. The classifier model is a two layer fully-connected neural network with 128 units at each layer; we repeat all experiments with 5 random seeds. As our goal here is to validate the consistency claims, we only take *i.i.d.* samples uniformly and do not perform any adaptive queries.

If a likelihood-free acquisition function is indeed equivalent to PI or EI, then we should expect it to become closer to the ground truth as the number of observations n increases. We report the quality of the acquisition functions, measured in L_1 error from the ground truth, in Figure 3 (more in Figure 8, Appendix B.1). For LFBO, both errors are roughly inversely proportional to n, whereas the same can only be said for BORE and PI; the error between BORE and EI remains large even with many samples. Therefore, the empirical results align with our statements in Proposition 3.2, Theorem 3.4, and Corollary 3.5.

5.2. Real-world Benchmarks

Setup. We empirically evaluate our method on various real-world optimization tasks, including hyperparameter optimization of neural networks (HPOBench, Klein & Hutter (2019)) and neural architecture search (NAS-Bench-201, Dong & Yang (2020)). We provide more details for these experiments in Appendix B.2.

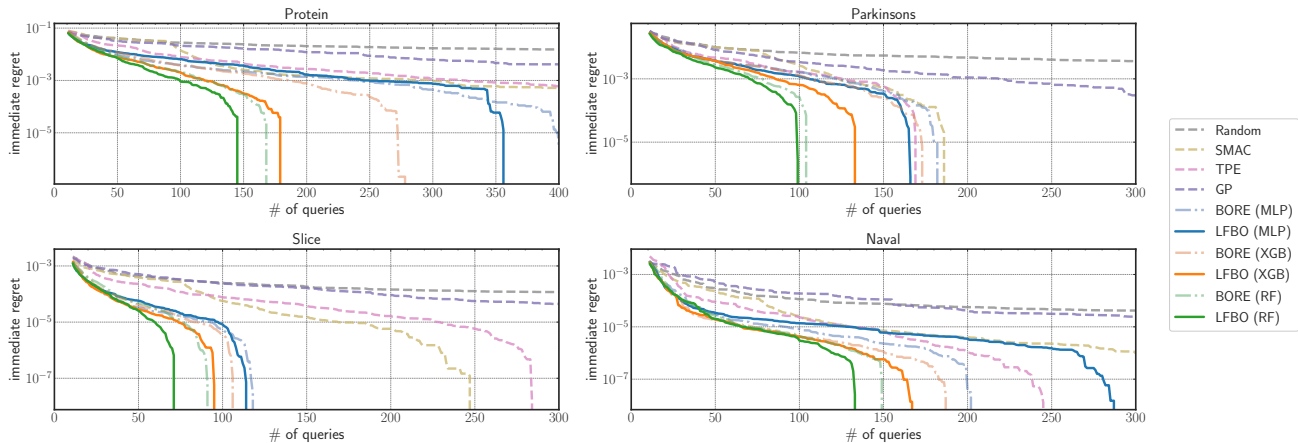In HPOBench, we aim to find the optimal hyperparameter

Figure 4. Immediate regret on the HPOBench neural network tuning problems. LFBO uses the utility function for EI. Each curve is the mean of results from 100 random seeds.
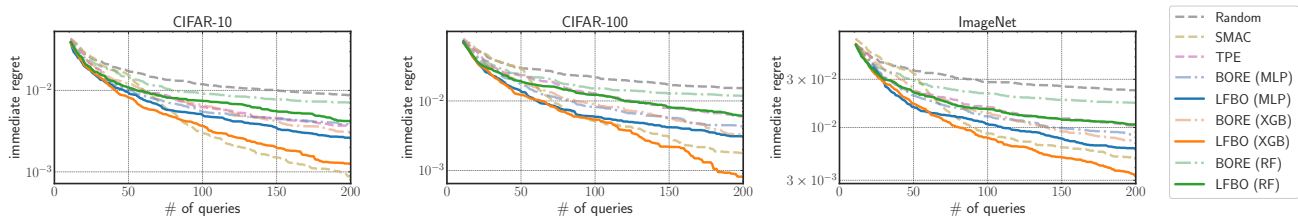

Figure 5. Immediate regret on the NAS-Bench-201 neural architecture search problems. LFBO uses the utility function for EI. Each curve is the mean of results from 100 random seeds.

configurations for training a two-layer feed-forward neural networks on four popular UCI datasets (Asuncion & Newman, 2007) for regression: protein structure (Protein), slice localization (Slice), naval propulsion (Naval) and parkinsons telemonitoring (Parkinsons). The configuration space includes batch size, initial learning rate, learning rate schedule, as well as layer-specific dropout rates, number of units and activation functions. For each dataset, HPOBench tabulates the mean squared errors of all $62,208$ configurations.

In NAS-Bench-201, the network architecture is constructed as a stack of neural cells, each represented as a directed acyclic graph (DAG), and we focus on the design of the neural cell, *i.e.*, assigning an operation to 6 edges of the DAG from a set of 5 operations. Similar to HPOBench, this dataset also provides a lookup table that tabulates the training results of all $15,625$ possible configurations for three datasets: CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009) and ImageNet-16 (Chrabaszcz et al., 2017).

In LFBO, we need to train a probabilistic classifier for candidate suggestion at each iteration. Here, we consider Multilayer perceptions (MLP), Random Forest (RF, Breiman (2001)), and gradient boosted trees (XGBoost, Chen & Guestrin (2016)). The MLP-based models are differentiable with respect to its inputs but requires special care for categorical variables (*e.g.*, with one-hot vectors). In contrast, the

tree-based ensemble classifiers can naturally deal with the discrete inputs but are non-differentiable. Despite this, for optimizing the acquisition functions, we use random search for all methods on all of the datasets, *i.e.*, first randomly sampling a large batch of candidates and then selecting the one with the maximum acquisition value in Equation 6.

We compare our approach with a comprehensive set of state-of-the-art black-box optimization methods, including BORE (Tiao et al., 2021), Tree-structured Parzen Estimator (TPE, Bergstra et al. (2011)), Sequential Model-based Algorithm Configuration (SMAC, Hutter et al. (2011)), GP-based Expected Improvement (Jones et al., 1998; Balandat et al., 2020) and Random Search (Bergstra & Bengio, 2012). Since LFBO with the utility function for PI would behave similarly to BORE, we consider LFBO with the utility function for EI in these experiments. Our implementation is based on an implementation of BORE, and we inherit all its default hyperparameters. Notably, we select the threshold τ to be the $(1 - \gamma)$-th quantile of the observed values, which includes less positive samples as γ becomes smaller. We select $\gamma = 0.33$[5], following (Tiao et al., 2021).

Results. Following (Tiao et al., 2021), we report the immediate regret (*i.e.* the absolute error between the global

---

[5]We perform a brief ablation study on this parameter in Appendix B.3, showing that LFBO (EI) is less sensitive to it.
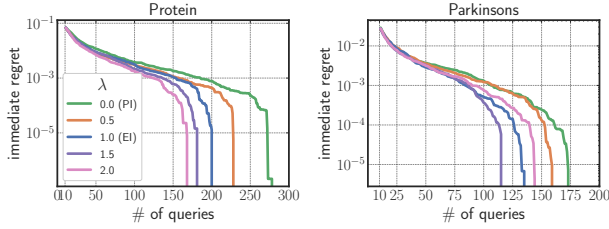
Figure 6. Results on two HPOBench problems with different utility functions, controlled by λ.

minimum of each benchmark and the lowest function value observed so far)[6] for quantitative evaluation of each method. The results for HPOBench and NAS-Bench-201 are summarized in Figure 4 and Figure 5 respectively, where the results are averaged across *100 random seeds*. We can see that our LFBO approach consistently achieves the best result across all the datasets, except on NAS-Bench-201-CIFAR-10 where the SMAC method is slightly better.

Notably, with all three classifier implementations (MLP, RF and XGBoost), LFBO (EI) consistently outperforms its BORE counterparts except for Naval with MLP, which demonstrates the effectiveness of choosing EI as the acquisition function and the value of enabling EI in the likelihood-free scenario. Moreover, we observe that GP-based EI methods (where we use Matern-Kernel with default hyperparameters values in the BoTorch[7] library) perform poorly on both benchmarks (omitted on NAS-Bench-201 where its performance is close to random search), possibly due to the complexity of attempting to learn a full distributional description of the underlying function. These results show the great potential of our generalized likelihood-free BO paradigm for strategic sequential decision making tasks.

Alternative utility functions. Since LFBO allows any nonnegative utility function, we may consider other alternatives to EI and PI. For example, we consider the following function: $u^\lambda(y; \tau) = (y - \tau)^\lambda$ if $y > \tau$; 0 otherwise. u^λ generalizes EI when $\lambda = 1$ and PI when $\lambda = 0$. Figure 6 illustrate the results on two HPOBench problems with $\lambda \in \{0.0, 0.5, 1.0, 1.5, 2.0\}$ and the XGBoost classifier. These results show that performance on these tasks increases as λ goes from 0 (PI) to 1 (EI), and that further improvements can be achieved by considering λ greater than

1; in this case, the utility with $\lambda = 1.5$ outperforms the EI one on both problems.

5.3. LFBO with Composite Objective Functions

Here, we consider the case of composite objective functions (Astudillo & Frazier, 2019; Maddox et al., 2021), where $g(\mathbf{x})$ is a known transformation of a vector-valued black-box function $h(\mathbf{x}) : \mathcal{X} \to \mathbb{R}^d$, *e.g.*,

$$g(\mathbf{x}) = -\|h(\mathbf{x}) - z_\star\|_2^2,$$

and $z_\star \in \mathbb{R}^d$ is a vector observed from the real world. Compared to standard BO approaches that do not exploit this information, a grey-box BO method that explicitly models h can make a much better sampling decision (Astudillo & Frazier, 2022). However, if we model $h(\mathbf{x})$ with GPs, then the EI acquisition function defined over the above g is no longer tractable, and traditional approaches would rely on closed-form but biased approximations (Uhrenholt & Jensen, 2019) or unbiased but slow Monte Carlo estimates (Astudillo & Frazier, 2019).

In LFBO, we can also exploit the composite nature of the objective function to obtain more efficient BO algorithms. Specifically, we may consider a composite neural network $C_\theta(\mathbf{x}) : \mathcal{X} \to \mathbb{R}$ parametrized by training parameters θ:

$$C_\theta(\mathbf{x}) = \frac{u(-\|h_\theta(\mathbf{x}) - z_\star\|_2^2; \tau)}{u(-\|h_\theta(\mathbf{x}) - z_\star\|_2^2; \tau) + 1},$$

where we use a neural network[8] for $h_\theta(\mathbf{x}) : \mathcal{X} \to \mathbb{R}^d$. As long as u allows automatic differentiation (*e.g.*, in the case of EI), we can apply backpropagation over θ to optimize it.

We consider a well-known test problem in the Bayesian calibration literature (Bliznyuk et al., 2008) that models the concentration of substances in a chemical accident. Our goal is to find the underlying parameters, such as the mass of pollutants, that minimizes the sum of squared errors with the real-world observation at 12 locations. We include more details in Appendix B.4.

Results of this experiment are in Figure 7. Our composite LFBO (EI) approach performs significantly better than regular GP and LFBO, while being on par with a GP approach based on composite EI (Astudillo & Frazier, 2019). Moreover, once our neural network has been trained, we do not require sampling from the surrogate to evaluate the composite acquisition function (unlike the GP counterpart which relies on Monte Carlo samples). Even with training taken into account, our composite LFBO method is much

---

[6]We note that here and in (Tiao et al., 2021), the actual output from the black-box function is a random value from several possible ones (simulating noise in the optimization process), whereas the global minimum considers the expectation over the values. Therefore, a negative immediate regret is possible (and we would use the corresponding neural network for deployment). The flat vertical lines in our plots mean that the average immediate regret (across all the seeds) has become negative.

[7]https://github.com/pytorch/botorch/blob/main/botorch/models/gp_regression.py

[8]While other models (such as trees) can be used here in principle as well, we focus on neural networks as gradient-based optimization of the likelihood-free objective function (training) is more straightforward.
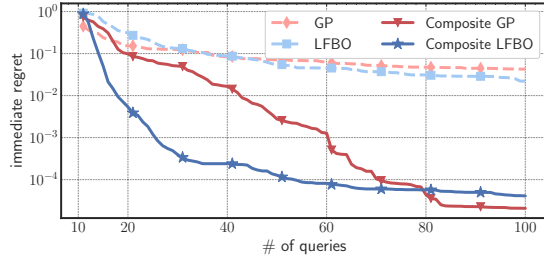
Figure 7. Results on composite objective function optimization. The utility function for EI is used by all the methods considered.

more efficient (around $15\times$ to $30\times$) when evaluated on the same hardware.

6. Discussions

We have introduced likelihood-free BO (LFBO) that extends BO to general surrogate models and utility functions. Motivated by likelihood-free inference, LFBO directly estimates the acquisition function from samples. With infinite query efforts in the search space, LFBO will converge to the desired expected utility function. LFBO can be reduced to learning the optimal classifier of a weighted classification problem. Empirical results demonstrate both the validity of our theory and the usefulness of LFBO (EI) empirically in black-box and composite function optimization settings.

Limitations While we observe some empirical successes with LFBO, we believe that it has certain limitations.

- First, because LFBO does not explicitly learn a distributional representation about our current knowledge of the black-box function, the behaviors of the algorithm (explore-exploit trade-off) may be less interpretable.

- Second, LFBO combines two separate steps (learning the surrogate model, and maximizing the acquisition function based on the learned surrogate model) into one classification problem, thus the trained model is tied to one specific acquisition function, while GP-based methods can be more flexible since the surrogate model and the acquisition function are decoupled.

- Third, LFBO has to choose a threshold below the current optimal observation, since otherwise no observations will fall into the positive class, and the model will not learn anything useful. This is also tied to the fact that LFBO models do not provide a probabilistic interpretation to the underlying function.

- Finally, there is a risk where the acquisition function from LFBO is over-confident, and would reduce exploration diversity (see the example on the Forrester function in Figure 11).

We believe that these could be mitigated with approaches that still provide a probabilistic interpretation to the black-box function, but are not necessarily have tractable acquisition functions like GPs; we can use LFBO as trying use a model to amortize the acquisition function (that is otherwise intractable), which could be faster to optimize than using Monte Carlo samples.

Future work Apart from applying LFBO to various application domains, there are several interesting avenues for future works:

- Extensions to more sophisticated BO paradigms, such as batch-based, multi-fidelity (Forrester et al., 2007), multi-objective (Hernández-Lobato et al., 2016), and complex composite functions (Astudillo & Frazier, 2021).

- Integration of LFBO with explicit and implicit surrogate functions as an amortization approach, which may accelerate approaches such as knowledge gradients (Frazier et al., 2008).

- LFBO with alternative choices of the scoring rule (defined via f), better heuristics of selecting the threshold τ, and additional techniques that bridge the gap between LFBO and GP-based BO.

- Exploring LFBO with other information acquired when the exact function value is unknown, such as ordering relationships (Christiano et al., 2017).

Acknowledgements

References

Astudillo, R. and Frazier, P. Bayesian optimization of composite functions. In *International Conference on Machine Learning*, pp. 354–363. PMLR, 2019.

Astudillo, R. and Frazier, P. Bayesian optimization of function networks. *Advances in Neural Information Processing Systems*, 34, 2021.

Astudillo, R. and Frazier, P. I. Thinking inside the box: A tutorial on grey-box bayesian optimization. *arXiv preprint arXiv:2201.00272*, 2022.

Asuncion, A. and Newman, D. UCI machine learning repository, 2007.

Attia, P. M., Grover, A., Jin, N., Severson, K. A., Markov, T. M., Liao, Y.-H., Chen, M. H., Cheong, B., Perkins, N., Yang, Z., et al. Closed-loop optimization of fast-charging protocols for batteries with machine learning. *Nature*, 578(7795):397–402, 2020.

Balandat, M., Karrer, B., Jiang, D., Daulton, S., Letham, B., Wilson, A. G., and Bakshy, E. Botorch: A framework for efficient monte-carlo bayesian optimization. *Advances in neural information processing systems*, 33, 2020.

Bergstra, J. and Bengio, Y. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, 2011.

Bergstra, J., Yamins, D., and Cox, D. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International conference on machine learning*, pp. 115–123. PMLR, 2013.

Bickel, S., Brückner, M., and Scheffer, T. Discriminative learning for differing training and test distributions. In *Proceedings of the 24th international conference on Machine learning*, pp. 81–88, 2007.

Bliznyuk, N., Ruppert, D., Shoemaker, C., Regis, R., Wild, S., and Mugunthan, P. Bayesian calibration and uncertainty analysis for computationally expensive models using optimization and radial basis function approximation. *Journal of Computational and Graphical Statistics*, 17 (2):270–294, 2008.

Breiman, L. Random forests. *Machine learning*, 45(1): 5–32, 2001.

Brochu, E., De Freitas, N., and Ghosh, A. Active preference learning with discrete choice data. In *NIPS*, pp. 409–416, 2007.

Buja, A., Stuetzle, W., and Shen, Y. Loss functions for binary class probability estimation and classification: Structure and applications. *Working draft, November*, 3, 2005.

Calandra, R., Seyfarth, A., Peters, J., and Deisenroth, M. P. Bayesian optimization for learning gaits under uncertainty. *Annals of Mathematics and Artificial Intelligence*, 76(1):5–23, 2016.

Chen, T. and Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

Chrabaszcz, P., Loshchilov, I., and Hutter, F. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Deshpande, S. and Kuleshov, V. Calibration improves bayesian optimization. *arXiv preprint arXiv:2112.04620*, 2021.

Diggle, P. J. and Gratton, R. J. Monte carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 46 (2):193–212, 1984.

Dong, X. and Yang, Y. Nas-bench-201: Extending the scope of reproducible neural architecture search. *arXiv preprint arXiv:2001.00326*, 2020.

Forrester, A. I., Sóbester, A., and Keane, A. J. Multi-fidelity optimization via surrogate modelling. *Proceedings of the royal society a: mathematical, physical and engineering sciences*, 463(2088):3251–3269, 2007.

Frazier, P. I., Powell, W. B., and Dayanik, S. A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization*, 47(5):2410–2439, 2008.

Garnett, R. *Bayesian Optimization*. Cambridge University Press, 2022. in preparation.

Gneiting, T. and Raftery, A. E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, March 2007. ISSN 0162-1459. doi: 10.1198/016214506000001437.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. Curran Associates, Inc., 2014.

Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., and Schölkopf, B. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4): 5, 2009.

Griffiths, R.-R. and Hernández-Lobato, J. M. Constrained bayesian optimization for automatic chemical design using variational autoencoders. *Chemical science*, 11(2): 577–586, 2020.

Gutmann, M. and Hyvarinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. *AISTATS*, 2010.

Hennig, P. and Schuler, C. J. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(6), 2012.

Hernández-Lobato, D., Hernandez-Lobato, J., Shah, A., and Adams, R. Predictive entropy search for multi-objective bayesian optimization. In *International Conference on Machine Learning*, pp. 1492–1501. PMLR, 2016.

Hernández-Lobato, J. M., Hoffman, M. W., and Ghahramani, Z. Predictive entropy search for efficient global optimization of black-box functions. *arXiv preprint arXiv:1406.2541*, 2014.

Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Hutter, F., Hoos, H. H., and Leyton-Brown, K. Sequential model-based optimization for general algorithm configuration. In *International conference on learning and intelligent optimization*, pp. 507–523. Springer, 2011.

Jones, D. R., Schonlau, M., and Welch, W. J. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

Kanamori, T., Suzuki, T., and Sugiyama, M. Theoretical analysis of density ratio estimation. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 93(4):787–798, 2010.

Klein, A. and Hutter, F. Tabular benchmarks for joint architecture and hyperparameter optimization. *arXiv preprint arXiv:1905.04970*, 2019.

Kleinegesse, S. and Gutmann, M. U. Efficient bayesian experimental design for implicit models. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 476–485. PMLR, 2019.

Koyama, Y., Sato, I., Sakamoto, D., and Igarashi, T. Sequential line search for efficient visual design optimization by crowds. *ACM Transactions on Graphics (TOG)*, 36(4): 1–11, 2017.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Kushner, H. J. A versatile stochastic model of a function of unknown and time varying form. *Journal of Mathematical Analysis and Applications*, 5(1):150–167, 1962.

Kushner, H. J. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. 1964.

Maddox, W. J., Balandat, M., Wilson, A. G., and Bakshy, E. Bayesian optimization with high-dimensional outputs. *Advances in Neural Information Processing Systems*, 34, 2021.

McIntire, M., Ratner, D., and Ermon, S. Sparse gaussian processes for bayesian optimization. In *UAI*, 2016.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 26*, pp. 3111–3119. Curran Associates, Inc., 2013.

Močkus, J. On bayesian methods for seeking the extremum. In *Optimization techniques IFIP technical conference*, pp. 400–404. Springer, 1975.

Mockus, J., Tiesis, V., and Zilinskas, A. The application of bayesian methods for seeking the extremum. *Towards global optimization*, 2(117-129):2, 1978.

Mohamed, S. and Lakshminarayanan, B. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*, October 2016.

Nguyen, X., Wainwright, M. J., and Jordan, M. I. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *arXiv preprint arXiv:0809.0853*, (11): 5847–5861, September 2008. doi: 10.1109/TIT.2010. 2068870.

Pearce, M. and Branke, J. Bayesian simulation optimization with input uncertainty. In *2017 Winter Simulation Conference (WSC)*, pp. 2268–2278. IEEE, 2017.

Perrone, V., Jenatton, R., Seeger, M., and Archambeau, C. Scalable hyperparameter transfer learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 6846–6856, 2018.

Rasmussen, C. E. Gaussian processes in machine learning. In *Summer school on machine learning*, pp. 63–71. Springer, 2003.

Reid, M., Williamson, R., et al. Information, divergence and risk for binary experiments. 2011.

Scott, W., Frazier, P., and Powell, W. The correlated knowledge gradient for simulation optimization of continuous parameters using gaussian process regression. *SIAM Journal on Optimization*, 21(3):996–1026, 2011.

Snoek, J., Larochelle, H., and Adams, R. P. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M., Prabhat, M., and Adams, R. Scalable bayesian optimization using deep neural networks. In *International conference on machine learning*, pp. 2171–2180. PMLR, 2015.

Sobester, A., Forrester, A., and Keane, A. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008.

Song, J. and Ermon, S. Bridging the gap between f-gans and wasserstein gans. In *International Conference on Machine Learning*, 2020.

Springenberg, J. T., Klein, A., Falkner, S., and Hutter, F. Bayesian optimization with robust bayesian neural networks. *Advances in neural information processing systems*, 29:4134–4142, 2016.

Sugiyama, M., Nakajima, S., Kashima, H., Von Buenau, P., and Kawanabe, M. Direct importance estimation with model selection and its application to covariate shift adaptation. In *NIPS*, volume 7, pp. 1433–1440. Citeseer, 2007.

Sugiyama, M., Suzuki, T., and Kanamori, T. *Density ratio estimation in machine learning*. Cambridge University Press, 2012.

Sugiyama, M., Kanamori, T., Suzuki, T., du Plessis, M. C., Liu, S., and Takeuchi, I. Density-difference estimation. *Neural Computation*, 25(10):2734–2775, 2013.

Swersky, K., Snoek, J., and Adams, R. P. Multi-task bayesian optimization. 2013.

Tiao, L. C., Klein, A., Seeger, M., Bonilla, E. V., Archambeau, C., and Ramos, F. Bore: Bayesian optimization by density-ratio estimation. *arXiv preprint arXiv:2102.09009*, 2021.

Titsias, M. Variational learning of inducing variables in sparse gaussian processes. In *Artificial intelligence and statistics*, pp. 567–574. PMLR, 2009.

Tran, D., Ranganath, R., and Blei, D. M. Hierarchical implicit models and likelihood-free variational inference. *arXiv preprint arXiv:1702.08896*, 2017.

Tran, K., Neiswanger, W., Yoon, J., Zhang, Q., Xing, E., and Ulissi, Z. W. Methods for comparing uncertainty quantifications for material property predictions. *Machine Learning: Science and Technology*, 1(2):025006, 2020.

Uehara, M., Sato, I., Suzuki, M., Nakayama, K., and Matsuo, Y. Generative adversarial nets from a density ratio estimation perspective. *arXiv preprint arXiv:1610.02920*, October 2016.

Uhrenholt, A. K. and Jensen, B. S. Efficient bayesian optimization for target vector estimation. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2661–2670. PMLR, 2019.

Wang, H., Zhang, X., and Ng, S. H. A nonparametric bayesian approach for simulation optimization with input uncertainty. *arXiv preprint arXiv:2008.02154*, 2020.

White, C., Neiswanger, W., and Savani, Y. Bananas: Bayesian optimization with neural architectures for neural architecture search. *arXiv preprint arXiv:1910.11858*, 1(2), 2019.

Williams, C. K. and Rasmussen, C. E. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

Wilson, J. T., Hutter, F., and Deisenroth, M. P. Maximizing acquisition functions for bayesian optimization. *arXiv preprint arXiv:1805.10196*, 2018.

Wistuba, M. and Grabocka, J. Few-shot bayesian optimization with deep kernel surrogates. *arXiv preprint arXiv:2101.07667*, 2021.

Yu, L., Song, J., and Ermon, S. Multi-Agent adversarial inverse reinforcement learning. *arXiv preprint arXiv:1907.13220*, July 2019.

Yu, L., Song, Y., Song, J., and Ermon, S. Training deep energy-based models with f-divergence minimization. In *International Conference on Machine Learning*, pp. 10957–10967. PMLR, 2020.

Zhang, D., Fu, J., Bengio, Y., and Courville, A. Unifying likelihood-free inference with black-box sequence design and beyond. *arXiv preprint arXiv:2110.03372*, 2021a.

Zhang, J., Bi, S., and Zhang, G. A scalable gradient free method for bayesian experimental design with implicit models. In *International Conference on Artificial Intelligence and Statistics*, pp. 3745–3753. PMLR, 2021b.

A. Proofs

A.1. Definitions and Assumptions

Definition A.1 (A valid sequential decision making process). Let $p_t(\mathbf{x}|\mathcal{D}_{t-1})$ denote the distribution over $\mathbf{x}$ at time t given observations $\mathcal{D}_{t-1} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{t-1}$ (which depends on the decision making process), and let $p(y|\mathbf{x})$ be the conditional distribution (which depends on the true black-box function g); then we define a sequential and Markovian decision making process as follows:

$$\mathbf{x}_t \sim p_t(\mathbf{x}|\mathcal{D}_{t-1}), y_t \sim p(y|\mathbf{x}), \mathcal{D}_t = \mathcal{D}_{t-1} \cup \{(\mathbf{x}_t, y_t)\}, \mathcal{D}_0 = \varnothing. \tag{8}$$

Define $p_t(\mathbf{x})$ as the marginal distribution of the selected $\mathbf{x}$ at time t; a sequential decision making process is valid as long as $\text{plim}_{t \to \infty} p_t(\mathbf{x})$ exists and is supported on the search space $\mathcal{X}$. We would commonly denote this limiting distribution as $p(\mathbf{x})$.

Remark A.2. An ϵ-greedy strategy generates a valid sequential decision making process (which would contain a mixture component that is the uniform distribution over $\mathcal{X}$). Moreover, in both expected utility and density ratios, the marginal distribution over $\mathbf{x}$ can be canceled out, so it does not affect the acquisition functions discussed in this paper.

The existence of a limiting "ground truth" distribution supported over the entire search space allows us to define to whose statistics our model estimators are converging to in probability.

Assumption A.3 (Consistency of statistical models). Let $\mathcal{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ be a set of n observations drawn from a valid sequential decision making process, whose limiting distribution is defined as $p(\mathbf{x}, y)$. For any statistical model that we consider, let $E_n(\mathcal{D}_n)$ be the estimator of a statistic E of $p(\mathbf{x}, y)$, then:

$$\text{plim}_{n \to \infty} E_n(\mathcal{D}_n) = E. \tag{9}$$

For example:

$$\text{plim}_{n \to \infty} p(y|\mathbf{x}, \mathcal{D}_n) = p(y|\mathbf{x}), \quad \forall \mathbf{x} \in \mathcal{X}, y \in \mathbb{R} \tag{10}$$

$$\text{plim}_{n \to \infty} p(\mathbf{x}|\mathcal{D}_n) = p(\mathbf{x}), \quad \forall \mathbf{x} \in \mathcal{X} \tag{11}$$

$$\text{plim}_{n \to \infty} p(\mathbf{x}|y > \tau, \mathcal{D}_n) = p(\mathbf{x}|y > \tau), \quad \forall \mathbf{x} \in \mathcal{X}, y \in \mathbb{R}, \tau \in \mathbb{R}. \tag{12}$$

A.2. Statements Regarding Density Ratios

These results are also discussed by this textbook (Garnett, 2022), albeit without an explicit proof.

Proposition 3.2. L^{DR} *is equivalent to* $\mathcal{L}^{\text{PI}}$.

Proof. From the consistency assumptions over $p(y|\mathbf{x}, \mathcal{D}_n)$, we have that $\forall \mathbf{x} \in \mathcal{X}, y \in \mathbb{R}$:

$$\text{plim}_{n \to \infty} \mathcal{L}^{\text{PI}}(\mathbf{x}, \mathcal{D}_n, \tau) = \text{plim}_{n \to \infty} \mathbb{E}_{p(y|\mathbf{x}, \mathcal{D}_n)}[\mathbb{I}(y > \tau)] \tag{13}$$

$$= \text{plim}_{n \to \infty} \int_{y=\tau}^{\infty} p(y|\mathbf{x}, \mathcal{D}_n) \mathrm{d}y \tag{14}$$

$$= \text{plim}_{n \to \infty} p(y > \tau|\mathbf{x}, \mathcal{D}_n) = p(y > \tau|\mathbf{x}). \tag{15}$$

From the consistency assumptions over $p(\mathbf{x}|y > \tau, \mathcal{D}_n)$ and $p(\mathbf{x}|\mathcal{D}_n)$, we have that $\forall \mathbf{x} \in \mathcal{X}, y \in \mathbb{R}$:

$$\gamma \text{plim}_{n \to \infty} L^{\text{DR}}(x; \mathcal{D}_n, \tau) = \gamma \text{plim}_{n \to \infty} \frac{p(\mathbf{x}|y > \tau, \mathcal{D})}{p(\mathbf{x}|\mathcal{D})} \tag{16}$$

$$= \gamma \frac{p(\mathbf{x}|y > \tau)}{p(\mathbf{x})} \tag{17}$$

$$= \frac{p(\mathbf{x}|y > \tau)p(y > \tau)}{p(\mathbf{x})} = p(y > \tau|\mathbf{x}), \tag{18}$$

where $\gamma := p(y > \tau)$ does not depend on $\mathbf{x}$, and we apply Slutsky's theorem in Equation 17. Therefore,

$$\operatorname*{plim}_{n \to \infty} L^{\mathrm{DR}}(\mathbf{x}; \mathcal{D}_n, \tau) \propto \operatorname*{plim}_{n \to \infty} \mathcal{L}^{\mathrm{PI}}(\mathbf{x}; \mathcal{D}_n, \tau) \tag{19}$$

which completes the proof. □

Corollary A.4. L^{DR} *is not equivalent to* $\mathcal{L}^{\mathrm{EI}}$.

Proof. Let $p(x) = \mathcal{N}(0, 1)$, $p(y|x) = \mathcal{N}(x, 1)$ defines a limit to the valid sequential decision process in Definition A.1, and let $\tau = 0$. Then applying the consistency assumptions over $p(y|x, \mathcal{D}_n)$:

$$\operatorname*{plim}_{n \to \infty} \mathcal{L}^{\mathrm{PI}}(x; \mathcal{D}_n, \tau) = \mathbb{E}_{p(y|x)}[\mathbb{I}(y > 0)] = \Phi(x), \tag{20}$$

where $\Phi(x)$ is the CDF of the standard Gaussian distribution. Meanwhile,

$$\operatorname*{plim}_{n \to \infty} \mathcal{L}^{\mathrm{EI}}(x; \mathcal{D}_n, \tau) = \mathbb{E}_{p(y|x)}[\max(y, 0)] = x\Phi(x) + \phi(x), \tag{21}$$

where $\phi(x)$ is the PDF of the standard Gaussian distribution. Applying Proposition 3.2, we have that:

$$\operatorname*{plim}_{n \to \infty} L^{\mathrm{DR}}(x; \mathcal{D}_n, \tau) = \operatorname*{plim}_{n \to \infty} \mathcal{L}^{\mathrm{PI}}(x; \mathcal{D}_n, \tau) \neq \operatorname*{plim}_{n \to \infty} \mathcal{L}^{\mathrm{EI}}(x; \mathcal{D}_n, \tau), \tag{22}$$

for some x, which completes the proof.

We note that the statement directly contradicts previous claims regarding the relationship between EI and DR (Bergstra et al., 2011; Tiao et al., 2021). The following claim restates the results in Bergstra et al. (2011); Tiao et al. (2021), which claim an equivalence between density (or likelihood) ratio acquisition functions and expected improvements[9]:

Claim A.5 ((Bergstra et al., 2011; Tiao et al., 2021)). Following notations in Tiao et al. (2021), let τ be the $(1-\gamma)$-th percentile of observed y values (*i.e.*, $\gamma = p(y > \tau)$), and let two densities $\ell(\mathbf{x}) := p(\mathbf{x}|y \leq \tau, \mathcal{D}_n)$ and $g(\mathbf{x}) := p(\mathbf{x}|y > \tau, \mathcal{D}_n)$ be the densities for $\mathbf{x}$ conditioned on y being less or greater than the threshold, respectively. Then:

$$\mathrm{EI}(\mathbf{x}, \tau) := \mathbb{E}_{p(y|\mathbf{x}, D_n)}[\max(y - \tau, 0)] \propto \frac{p(\mathbf{x}|y > \tau, \mathcal{D}_n)}{(1 - \gamma)p(\mathbf{x}|y \leq \tau, \mathcal{D}_n) + \gamma p(\mathbf{x}|y > \tau, \mathcal{D}_n)}. \tag{23}$$

Counter claim. We will now go through the "proof" of the claim and identify where the argument does not hold.

First we have that:

$$\mathrm{EI}(\mathbf{x}, \tau) = \mathbb{E}_{p(y|\mathbf{x}, D_n)}[\max(y - \tau, 0)] = \int_\tau^\infty \max(y - \tau, 0)p(y|\mathbf{x}, D_n)\mathrm{d}y \tag{24}$$

$$= \frac{\int_\tau^\infty \max(y - \tau, 0)p(y|D_n)p(\mathbf{x}|y, D_n)\mathrm{d}y}{p(\mathbf{x}|D_n)}, \tag{25}$$

where we applied Bayes' rule in Equation 25. Then, the denominator evaluates as:

$$p(\mathbf{x}|D_n) = p(\mathbf{x}|y \leq \tau, D_n)p(y \leq \tau|D_n) + p(\mathbf{x}|y > \tau, D_n)p(y > \tau|D_n) \tag{26}$$

$$= \ell(\mathbf{x})p(y \leq \tau|D_n) + g(\mathbf{x})p(y > \tau|D_n) = (1 - \gamma)\ell(\mathbf{x}) + (1 - \gamma)g(\mathbf{x}). \tag{27}$$

where we use the definition that $\gamma := p(y > \tau|D_n)$. Finally, Bergstra et al. (2011); Tiao et al. (2021) claims that the numerator in Equation 25 evaluates to:

$$\int_\tau^\infty \max(y - \tau, 0)p(y|D_n)p(\mathbf{x}|y, D_n)\mathrm{d}y = g(\mathbf{x}) \int_\tau^\infty \max(y - \tau, 0)p(y|D_n)\mathrm{d}y \tag{28}$$

---

[9]We note that Bergstra et al. (2011); Tiao et al. (2021) minimizes the BO objective in their formulation, whereas we paraphrase the claim to adapt to maximization.

which is $g(\mathbf{x})$ times a value independent of $\mathbf{x}$. Dividing Equation 28 with Equation 27 and using Equation 25 should recover the result. However, in Equation 28, $g(\mathbf{x}) := p(\mathbf{x}|y > \tau)$ is directly taken out of the integral, as Bergstra et al. (2011) assumed it is independent of y as long as $y > \tau$. From the definition of conditional probability:

$$g(\mathbf{x}) := p(\mathbf{x}|y > \tau, D_n) = \frac{\int_\tau^\infty p(\mathbf{x}, y|D_n)\mathrm{d}y}{\int_\tau^\infty p(y|D_n)\mathrm{d}y} \neq p(\mathbf{x}|y, D_n), \tag{29}$$

so Equation 28 does not hold immediately. Thus, we have illustrated that the above density ratio may not be proportional to expected improvement. □

A.3. Statements Regarding Likelihood-free BO

Lemma 3.3 (Variational representation of an integral)**.** *For all non-negative utility functions* $u : \mathbb{R} \times \mathcal{T} \to [0, \infty)$, *and for all strictly convex functions* $f : [0, \infty) \to \mathbb{R}$ *with third order derivatives,*

$$\mathbb{E}_{p(y|\mathbf{x})}[u(y; \tau)] = \underset{s \in [0,\infty)}{\arg\max} \, \mathbb{E}_{p(y|\mathbf{x})}[u(y; \tau)f'(s)] - f^\star(f'(s)),$$

where $f^\star$ *is the convex conjugate of* f, *and the maximization is performed over the* $s \in [0, \infty)$ *(the domain of* f).

Proof. From the definition of $f^\star$, we have that:

$$f^\star(f'(s)) = sf'(s) - f(s)$$
$$(f^\star \circ f')'(s) = f'(s) + sf''(s) - f'(s) = sf''(s)$$

The derivative of $\mathbb{E}_{p(y|\mathbf{x})}[u(y, \tau)f'(s)] - f^\star(f'(s))$ is thus:

$$\mathbb{E}_{p(y|\mathbf{x})}[u(y, \tau)]f''(s) - sf''(s); \tag{30}$$

since f is strictly convex, $f''(s) > 0$, and Equation 30 is zero if and only if $s = \mathbb{E}_{p(y|\mathbf{x})}[u(y, \tau)]$. Furthermore, the derivative of Equation 30 is

$$\mathbb{E}_{p(y|\mathbf{x})}[u(y, \tau)]f'''(s) - sf'''(s) - f''(s), \tag{31}$$

which is negative when $s = \mathbb{E}_{p(y|\mathbf{x})}[u(y, \tau)]$. Under this value of s, the first order derivative is zero and the second order derivative is negative, so this value of s is a unique maximizer. □

Theorem 3.4. *For* u *and* f *as defined in Lemma 3.3,* $\forall \mathbf{x} \in \mathcal{X}$ *and* $\forall \tau \in \mathcal{T}$, *let* $L_{\mathrm{LFBO}}^{(u)}(\mathbf{x}; \mathcal{D}_n, \tau) = \hat{S}_{\mathcal{D}_n, \tau}(\mathbf{x})$, *where*

$$\hat{S}_{\mathcal{D}_n, \tau} = \underset{S : \mathcal{X} \to \mathbb{R}}{\arg\max} \, \mathbb{E}_{\mathcal{D}_n}[u(y; \tau)f'(S(\mathbf{x})) - f^\star(f'(S(\mathbf{x})))].$$

Then $L_{\mathrm{LFBO}}^{(u)}$ *is equivalent to* $\mathcal{L}^{(u)}$ *in Equation 2.*

Proof. From Lemma 3.3, we have that for all $\mathbf{x}$ queried in $\mathcal{D}_n$,

$$\left(\underset{S : \mathcal{X} \to \mathbb{R}}{\arg\max} \, \mathbb{E}_{\mathcal{D}_n}[u(y; \tau)f'(S(\mathbf{x}')) - f^\star(f'(S(\mathbf{x}')))] \right)(\mathbf{x}) = \mathbb{E}_{p(y|\mathbf{x})}[u(y; \tau)].$$

As $n \to \infty$, $\mathcal{D}_n$ converges to a probability distribution $p(\mathbf{x})$ that is fully supported over $\mathcal{X}$. Let us denote the limiting distribution of $\mathcal{D}_n$ as $p(\mathbf{x})$, and define $p(\mathbf{y}|\mathbf{x})$ as the conditional distribution from the observation process. Applying the consistency assumption to $\hat{S}_{\mathcal{D}_n, \tau}$, we have that $\forall \mathbf{x} \in \mathcal{X}, \tau \in \mathcal{T}$:

$$\underset{n \to \infty}{\mathrm{plim}} \, L^{\mathrm{LF}(u)}(\mathbf{x}, \mathcal{D}_n, \tau) := \underset{n \to \infty}{\mathrm{plim}} \, \hat{S}_{\mathcal{D}_n, \tau}(\mathbf{x})$$

$$:= \underset{n \to \infty}{\mathrm{plim}} \left(\underset{S : \mathcal{X} \to \mathbb{R}}{\arg\max} \, \mathbb{E}_{\mathcal{D}_n}[u(y; \tau)f'(S(\mathbf{x}')) - f^\star(f'(S(\mathbf{x}')))] \right)(\mathbf{x}) \tag{32}$$

$$= \left(\underset{S : \mathcal{X} \to \mathbb{R}}{\arg\max} \, \mathbb{E}_{p(y|\mathbf{x})p(\mathbf{x}')}[u(y; \tau)f'(S(\mathbf{x}')) - f^\star(f'(S(\mathbf{x}')))] \right)(\mathbf{x}) = \mathbb{E}_{p(y|\mathbf{x})}[u(y; \tau)]. \tag{33}$$

Applying the consistency assumption to $p(y|\mathbf{x}, \mathcal{D}_n)$, we have that:

$$\plim_{n\to\infty} \mathcal{L}^{(u)}(\mathbf{x}, \mathcal{D}_n, \tau) = \mathbb{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[u(y;\tau)] = \mathbb{E}_{p(y|\mathbf{x})}[u(y;\tau)], \tag{34}$$

completing the proof. □

In the following, we show that the same expected utility function value can be achieved by different Gaussian distributions.

Proposition A.6 (Different Gaussian distributions can have the same expected utility). *Let $u(y;\tau)$ be non-negative, continuous and non-decreasing in the y argument for a given τ. For any Gaussian $p_1(y) = \mathcal{N}(\mu_1, \sigma_1^2)$, there exists a Gaussian $p_2(y) = \mathcal{N}(\mu_2, \sigma_2^2)$ such that $p_1 \neq p_2$ and*

$$\mathbb{E}_{p_1(y)}[u(y;\tau)] = \mathbb{E}_{p_2(y)}[u(y;\tau)].$$

Proof. Let $U := \mathbb{E}_{p_1(y)}[u(y;\tau)]$, and choose any fixed σ_2 smaller than σ_1. We have that:

$$\lim_{\mu_2\to\infty} \mathbb{E}_{p_2(y)}[u(y;\tau)] = \max_y u(y;\tau) \tag{35}$$

$$\lim_{\mu_2\to-\infty} \mathbb{E}_{p_2(y)}[u(y;\tau)] = \min_y u(y;\tau). \tag{36}$$

We split $u(y;\tau)$ into two cases:

- If $\max_y u(y;\tau) = \min_y u(y;\tau)$, then the desired condition is trivially satisfied.

- Otherwise, $U \in (\min_y u(y;\tau), \max_y u(y;\tau))$; since $\mathbb{E}_{p_2(y)}[u(y;\tau)]$ is a continous function over μ_2 when σ_2 is fixed, then from the intermediate value theorem, there must exist some μ_2 such that $U = \mathbb{E}_{p_2(y)}[u(y;\tau)]$.

Thus, there exists a different Gaussian with the same expected utility. □

B. Experimental Details

B.1. Synthetic Evaluation of LFBO and BORE (Section 5.1)

Our classifier model is a two layer fully-connected neural network with 128 units at each layer; we repeat all experiments with 5 random seeds. For each evaluation, we optimize the model with batch gradient descent for 1000 epochs, using an Adam optimizer with learning rate 0.01 and weight decay 10^{-6}. We illustrate the learned acquisition functions for one random seed and compare it with the ground truth ones in Figure 8. We placed a non-decreasing transformation over the BORE result to make it directly comparable to PI; suppose the density ratio predictor of BORE is C_{BORE}, we apply the following transformation (and implement a numerically stable version of it):

$$C_{\mathrm{BORE}} \mapsto \frac{C_{\mathrm{BORE}}}{C_{\mathrm{BORE}} + \frac{1-\gamma}{\gamma}}, \tag{37}$$

where $\gamma = p(y > \tau|\mathcal{D}_n)$. This does not modify the optimum of BORE, and is only applied when we would compare against the ground truth acquisition functions.
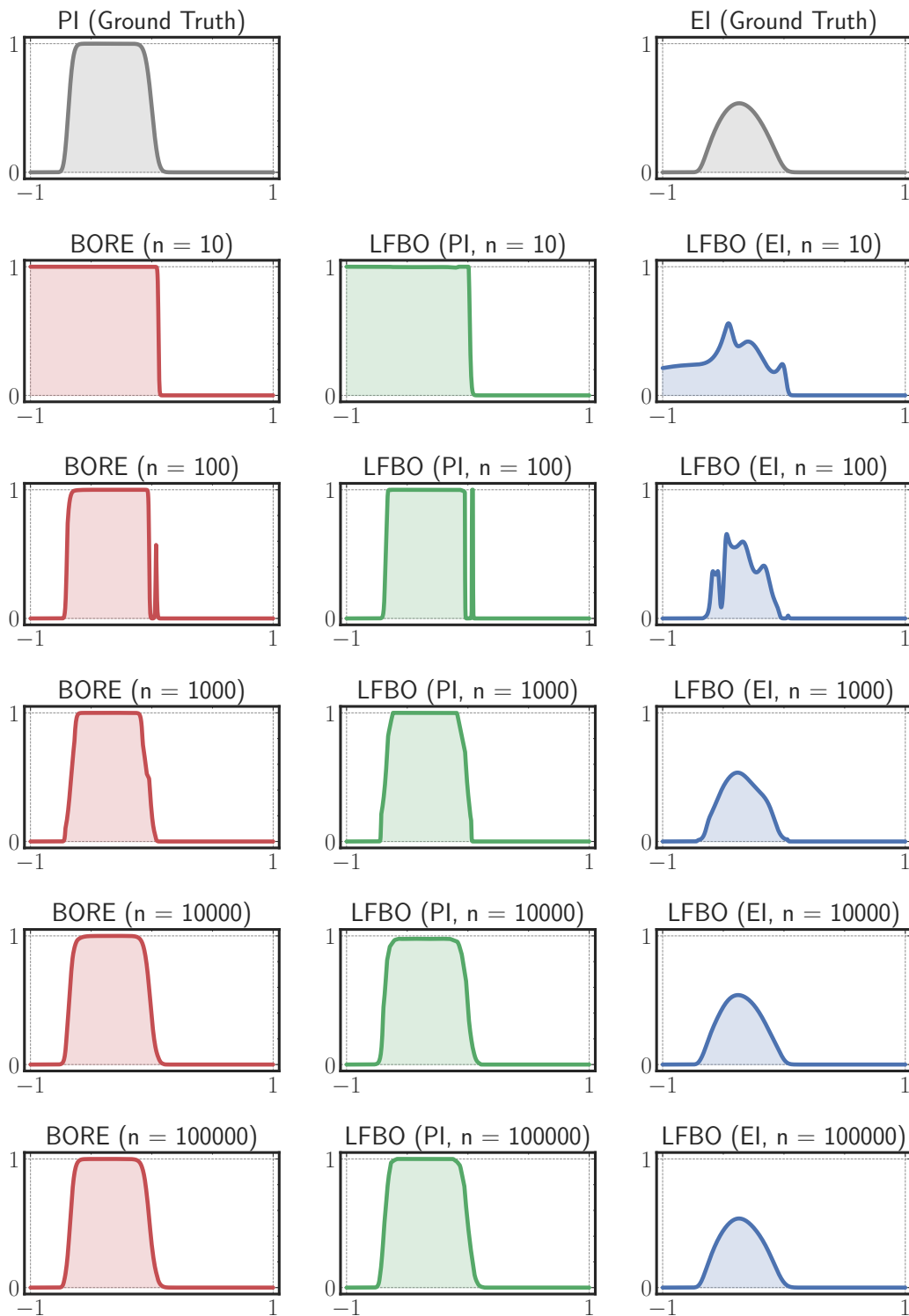
Figure 8. Acquisition functions for the synthetic problem. (*Top row*) ground truth PI and EI values; (*bottom rows*) acquisition functions obtained by BORE (*left column*), LFBO (PI, *middle column*), and LFBO (EI, *right column*) with the number of observations $n \in \{10, 10^2, 10^3, 10^4, 10^5, 10^6\}$. The acquisition functions are closer to the ground truth with more observations.

B.2. Hyperparameters for Real-world Benchmarks (Section 5.2)

We provide a description for the hyperparameter search space of the HPOBench (Klein & Hutter, 2019) and NAS-Bench-201 (Dong & Yang, 2020) problems in Table 1.

	Hyperparameter	Choices
HPOBench	Initial LR	$\{0.0005, 0.001, 0.005, 0.01, 0.05, 0.1\}$
	Batch Size	$\{8, 16, 32, 64\}$
	LR Schedule	$\{$cosine, fix$\}$
	Activation of Layer 1	$\{$relu, tanh$\}$
	Activation of Layer 2	$\{$relu, tanh$\}$
	Width of Layer 1	$\{16, 32, 64, 128, 256, 512\}$
	Width of Layer 2	$\{16, 32, 64, 128, 256, 512\}$
	Dropout rate of Layer 1	$\{0.0, 0.3, 0.6\}$
	Dropout rate of Layer 2	$\{0.0, 0.3, 0.6\}$
NAS-Bench-201	Arc-0 Operation	$\{$zeroize, skip-connect, conv-1 $\times$ 1, conv-3 $\times$ 3, avg-pool-3 $\times$ 3$\}$
	Arc-1 Operation	$\{$zeroize, skip-connect, conv-1 $\times$ 1, conv-3 $\times$ 3, avg-pool-3 $\times$ 3$\}$
	Arc-2 Operation	$\{$zeroize, skip-connect, conv-1 $\times$ 1, conv-3 $\times$ 3, avg-pool-3 $\times$ 3$\}$
	Arc-3 Operation	$\{$zeroize, skip-connect, conv-1 $\times$ 1, conv-3 $\times$ 3, avg-pool-3 $\times$ 3$\}$
	Arc-4 Operation	$\{$zeroize, skip-connect, conv-1 $\times$ 1, conv-3 $\times$ 3, avg-pool-3 $\times$ 3$\}$
	Arc-5 Operation	$\{$zeroize, skip-connect, conv-1 $\times$ 1, conv-3 $\times$ 3, avg-pool-3 $\times$ 3$\}$

Table 1. Configuration spaces for HPOBench and NAS-Bench-201 problems.

We follow (Tiao et al., 2021) to use Multi-layer perceptions (MLP), Random Forest (RF, Breiman (2001)), and gradient boosted trees (XGBoost, Chen & Guestrin (2016)) to implement the probabilistic classifier. Detailed hyperparameters are summarized below:

MLP: https://keras.io/
- Number of hidden layers: 2
- Number of hidden units: 32
- Activation function: ReLU
- Optimizer: Adam (with default parameters here)
- Batch size: 64

Random Forest: https://scikit-learn.org/
- Number of trees in the forest: 1000
- Minimum number of samples required to split an internal node: 2
- Maximum depth of the tree: None, nodes are expanded until all leaves contain less than 2 samples.
- Minimum number of samples required to be at a leaf node: 1
- More parameter values can be found here.

XGBoost: https://xgboost.readthedocs.io/
- Number of boosting rounds: 100
- Minimum sum of instance weight (hessian) needed in a child: 1
- Boosting learning rate: 0.3
- More parameter values can be found here.

B.3. Ablation Study over the Threshold

Both BORE and LFBO (EI) include a hyperparameter γ, which includes less positive samples when smaller and may affect optimization performance empirically. We use the XGBoost classifier on two HPOBench problems with different values of γ and plot the results in Figure 9. These results show that while we observe a performance drop with a larger γ in both cases, LFBO (EI) is less sensitive to the choice of γ, which is reasonable since it take into account how much the observed value exceeds the threshold.
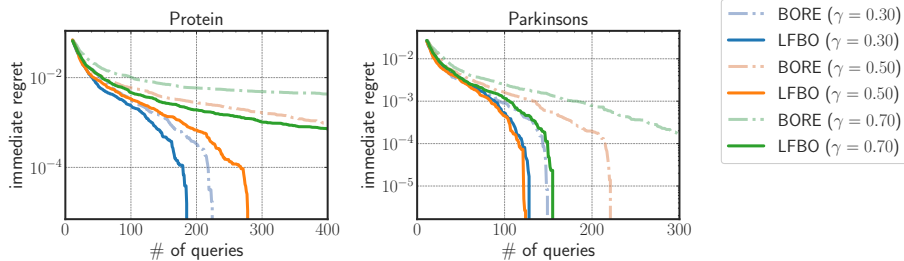
Figure 9. Results on two HPOBench problems with different thresholds. LFBO uses the utility function for EI.
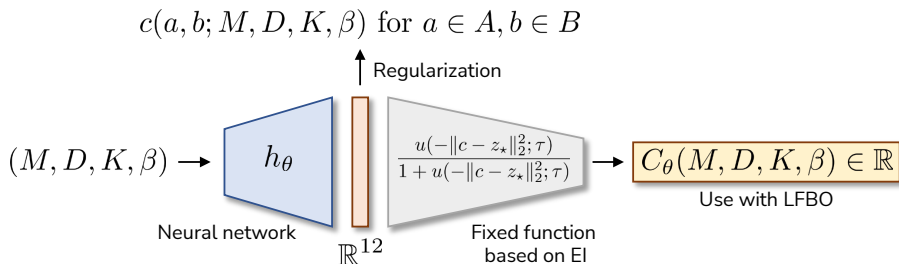

Figure 10. Structure for the classifier model in the composite case. We regularize the intermediate layer such that it is close to the real black-box outputs from h.

B.4. LFBO for Composite Functions

LFBO model and objectives. Our composite neural network is $C_\theta(\mathbf{x}) : \mathcal{X} \to \mathbb{R}$ parametrized by θ:

$$C_\theta(\mathbf{x}) = \frac{u(-\|h_\theta(\mathbf{x}) - z_\star\|_2^2; \tau)}{u(-\|h_\theta(\mathbf{x}) - z_\star\|_2^2; \tau) + 1},$$

where $h_\theta(\mathbf{x}) : \mathcal{X} \to \mathbb{R}^d$ is a neural network with the training parameters θ. Essentially, C_θ is a neural network that contains a special final layer that incorporates the structure of the objective function (see Figure 10). We also considered replacing the final layer with another neural network, but observed worse performance; this is not hard to expect since we would require some data to learn a suitable mapping in this case.

We consider training $C_\theta(\mathbf{x})$ with our LFBO objective plus a regularization term that encourages $h_\theta(\mathbf{x})$ (the neural network) to produce similar outputs to $h(\mathbf{x})$ (the black-box); for EI, this leads to the following objective function:

$$\frac{|\mathcal{D}_n^+|}{|\mathcal{D}_n|} \mathbb{E}_{\mathcal{D}_n^+}[(-\|\mathbf{y} - \mathbf{z}_\star\|_2^2 - \tau) \log C_\theta(\mathbf{x})] + \mathbb{E}_{\mathcal{D}_n}[\log(1 - C_\theta(\mathbf{x}))] + \mathbb{E}_{(\mathbf{x},\mathbf{y}) \sim \mathcal{D}_n}[-\|h_\theta(\mathbf{x}) - \mathbf{y}\|_2^2] \tag{38}$$

where $\mathcal{D}_n = \{(\mathbf{x}, \mathbf{y})\}_{i=1}^n$ contains observations from the black-box $h(\mathbf{x})$, and $\mathcal{D}_n^+$ is the subset of $\mathcal{D}_n$ where $-\|\mathbf{y} - \mathbf{z}_\star\|_2^2 > \tau$. We also tried different weights between the LFBO objective and the regularization term, which gave similar results.

Test problem. We evaluate our methods on the environmental model function (Bliznyuk et al., 2008), which models the spilling of pollutants at two sites into a long and narrow holding channel in a chemical accident. The model is:

$$c(a, b; M, D, K, \xi) = \frac{M}{\sqrt{4\pi D b}} \exp\left(\frac{-a^2}{4Db}\right) + \frac{\mathbb{I}(b - \xi > 0)M}{\sqrt{4\pi D(b - b_0)}} \exp\left(\frac{-(a - K)^2}{4D(b - \xi)}\right) \tag{39}$$
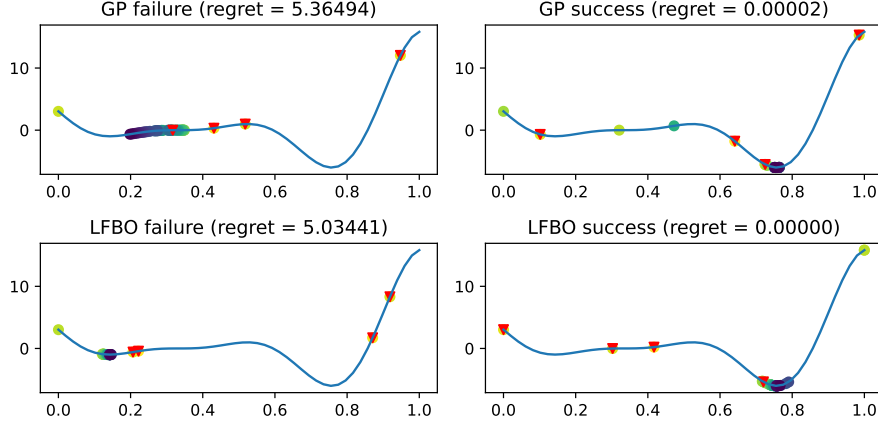
Figure 11. Success and failure cases for LFBO and GP (with EI) on minimizing the Forrester function (Sobester et al., 2008). The red triangle points indicate the initial random observations, and the subsequent observations are denoted by color circles (from bright to dark).

where M is the mass of pollutant spilled at each location, D is the diffusion rate in the channel, K is the location of the second spill, and ξ is the time of the second spill. The objective function is:

$$\sum_{(a,b) \in A \times B} (c(a,b; M, D, K, \xi) - c(a,b; M_0, D_0, K_0; \xi_0))^2 \tag{40}$$

where $A = \{0, 1, 2.5\}$, $B = \{15, 30, 45, 60\}$ are grid locations, and (M_0, D_0, K_0, ξ_0) are the underlying true values of these parameters. This function is a grey-box function in the sense that $c(a,b; M, D, K, \xi)$ for each $(a,b) \in A \times B$ is a black-box function.

Implementation and hyperparameters. We use our own implementation for GP (EI), LFBO (EI) and composite LFBO (EI), and use the official implementation[10] (with minor modifications for bug fixes) for composite GP (EI). We note that the regrets for composite GP (EI) appear larger than the ones reported in (Astudillo & Frazier, 2019) because of the evaluated statistics. Astudillo & Frazier (2019) use average of log10 regret, whereas we use log10 of average regret; since log10 is concave, our statistics will be larger than theirs (which would favor higher variance across random seeds).

For all the methods, we first take 10 random samples and then proceed with the Bayesian optimization algorithm; Figure 7 plots the immediate regret *after* the initial random samples have been collected. For composite LFBO (EI), we select τ to be the 10-th percentile of the existing observed $g(\mathbf{x})$ values. Our $h_\theta(\mathbf{x})$ is a two layer fully connected neural network with 64 units in each layer; we find that the results are insensitive to slight changes to the neural network architecture.

C. Additional Experimental Results

C.1. LFBO on Additional Continuous Functions

Apart from the composite function, we further evaluate LFBO on some additional continuous functions, such as the Forrester function (example from in Figure 11; we observe both successful and unsuccessful optimization results for both LFBO and GP, which depends on the initial random queries. Failures are mostly due to initial samples that causes the model to be overconfident about the global valley between $[0.6, 0.8]$, and even GPs cannot avoid them (Deshpande & Kuleshov, 2021) (these occur roughly 25% of the time for both methods). Improving LFBO on these failure cases is an interesting future work.

C.2. Runtime Comparisons between LFBO and GP-based Methods

We perform these comparisons on the same machine. On NAS-Bench-201's CIFAR-10 dataset, the average time for LFBO to finish 200 steps of BO is around 110 seconds, while the average time for GP-EI (implemented with the BoTorch library)

---

[10]https://github.com/RaulAstudillo06/BOCF/blob/master/test_4b.py

to finish 200 steps of BO is around 600 seconds. Thus LFBO is $5\times$ faster than the GP in this setting. We also find that model fitting and acquisition function optimization takes roughly the same time with LFBO, but the latter was much slower with GPs. Although both methods require some training time given new observations, the inference time of LFBO is much faster, since basic GP requires $\mathcal{O}(n^3)$ computational complexity to perform posterior inference, while in LFBO the model size (and by extension inference time) is constant. While sparse GPs can reduce inference time to the order of $\mathcal{O}(n)$, they can still be more expensive than LFBO (as n becomes very large).

On Bliznyuk, composite EI with GP (using the official implementation based on GPyTorch) is much slower (where 40 iterations took around 1200s), around $30\times$ slower than LFBO (which took around 40s). Even with a more efficient implementation using multi-task GPs (Maddox et al., 2021)[11], the 40 iterations took around 640s on the same CPU, which was around $15\times$ slower. We believe this is probably because the model has to take additional Monte Carlo samples as composite EI is no longer analytically tractable. While this gap could be smaller if further optimizations over the GP implementations are made, we believe that this demonstrates the efficiency of LFBO methods even with additional training.

---

[11]https://botorch.org/tutorials/composite_bo_with_hogp