Bayesian Optimization using Pseudo-Points

Chao Qian[1*], **Hang Xiong**[2] and **Ke Xue**[1]
[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
[2]University of Science and Technology of China, Hefei 230027, China
{qianc, xuek}@lamda.nju.edu.cn, flybear@mail.ustc.edu.cn

Abstract

Bayesian optimization (BO) is a popular approach for expensive black-box optimization, with applications including parameter tuning, experimental design, and robotics. BO usually models the objective function by a Gaussian process (GP), and iteratively samples the next data point by maximizing an acquisition function. In this paper, we propose a new general framework for BO by generating pseudo-points (i.e., data points whose objective values are not evaluated) to improve the GP model. With the classic acquisition function, i.e., upper confidence bound (UCB), we prove that the cumulative regret can be generally upper bounded. Experiments using UCB and other acquisition functions, i.e., probability of improvement (PI) and expectation of improvement (EI), on synthetic as well as real-world problems clearly show the advantage of generating pseudo-points.

1 Introduction

One often needs to solve an optimization problem: $\boldsymbol{x}^* \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x})$, where $\mathcal{X} \subseteq \mathbb{R}^d$ is the solution space, $f : \mathcal{X} \to \mathbb{R}$ is the objective function, and $\boldsymbol{x}^*$ is an optimal solution. Usually, it is assumed that f has a known mathematical expression, is convex, or cheap to evaluate at least. Increasing evidences, however, show that f may not satisfy these assumptions, but is an expensive black-box model [Brochu *et al.*, 2010]. That is, f can be non-convex, or even the closed-form expression of f is unknown; meanwhile, evaluating f can be noisy and computationally very expensive.

Expensive black-box optimization is involved in many real-world decision making problems. For example, in machine learning, one has to tune hyper-parameters to maximize the performance of a learning algorithm [Snoek *et al.*, 2012]; in physical experiments, one needs to set proper parameters of the experimental environment to obtain an ideal product [Brochu *et al.*, 2010]. More applications can been found in robotic control [Martinez-Cantin *et al.*, 2007], computer vision [Denil *et al.*, 2012], sensor placing [Garnett *et al.*, 2010], and analog circuit design [Lyu *et al.*, 2018].

BO [Mockus, 1994] has been a type of powerful algorithm to solve expensive black-box optimization problems. The main idea is to build a model, usually by a GP, for the objective f based on the observation data, and then sample the next data point by maximizing an acquisition function. Many BO algorithms have been proposed, with the goal of reaching the optima using as few objective evaluations as possible.

Most existing works focus on designing effective acquisition functions, e.g., PI [Kushner, 1964], EI [Jones *et al.*, 1998], and UCB [Srinivas *et al.*, 2012]. Recently, Wang et al. [2016] proposed the EST function by directly estimating $\boldsymbol{x}^*$, which automatically and adaptively trades off exploration and exploitation in PI and UCB. Another major type of acquisition function is based on information entropy, including entropy search (ES) [Hennig and Schuler, 2012], predictive ES [Hernández-Lobato *et al.*, 2014], max-value ES [Wang and Jegelka, 2017], FITBO [Ru *et al.*, 2018], etc. As BO is a sequential algorithm, some parallelization techniques have been introduced for acceleration, e.g., [Azimi *et al.*, 2010; Desautels *et al.*, 2014; Shah and Ghahramani, 2015; González *et al.*, 2016]. There is also a sequence of works addressing the difficulty of BO for high-dimensional optimization, e.g., [Wang *et al.*, 2013; Kandasamy *et al.*, 2015; Wang *et al.*, 2017; Mutny and Krause, 2018].

For any BO algorithm with a specific acquisition function, the GP model becomes increasingly accurate with the observation data augmenting. However, the number of data points to be evaluated is often limited due to the expensive objective evaluation. In this paper, we propose a general framework for BO by generating pseudo-points to improve the GP model. That is, before maximizing the acquisition function to select the next point in each iteration, some pseudo-points are generated and added to update the GP model. The pseudo-points are neighbors of the observed data points, and take the same function values as the observed ones. Without increasing the evaluation cost, the generation of pseudo-points can reduce the variance of the GP model, while introducing little accuracy loss under the Lipschitz assumption. This framework is briefly called BO-PP.

Theoretically, we study the performance of BO-PP w.r.t. the acquisition function UCB, called UCB-PP. We prove a

*This work was supported by the Fundamental Research Funds for the Central Universities (14380004) and the project of HUAWEI-LAMDA Joint Laboratory of Artificial Intelligence.

Algorithm 1 BO Framework

Input: iteration budget T
Process:
1: let $D_0 = \emptyset$;
2: **for** $t = 1 : T$ **do**
3: $\quad \boldsymbol{x}_t = \arg\max_{\boldsymbol{x} \in \mathcal{X}} acq(\boldsymbol{x})$;
4: $\quad$ evaluate f at $\boldsymbol{x}_t$ to obtain y_t;
5: $\quad$ augment the data $D_t = D_{t-1} \cup \{(\boldsymbol{x}_t, y_t)\}$ and update the GP model
6: **end for**

general upper bound of UCB-PP on the cumulative regret, i.e., $\sum_{t=1}^{T}(f(\boldsymbol{x}^*) - f(\boldsymbol{x}_t))$, where $\boldsymbol{x}_t$ denotes the sampled point in the t-th iteration. It is shown to be a generalization of the known bound [Srinivas *et al.*, 2012] of UCB. Empirically, we compare BO-PP with BO on synthetic benchmark functions as well as real-world optimization problems. The acquisition functions UCB, PI and EI are used. The results clearly show the superior performance of BO-PP.

2 Background

The general framework of BO is shown in Algorithm 1. It sequentially optimizes a given objective function $f(\boldsymbol{x})$ with assumptions on a prior distribution, i.e., a probabilistic model, over $f(\boldsymbol{x})$. In each iteration, BO selects a point $\boldsymbol{x}$ by maximizing an acquisition function $acq(\cdot)$, evaluates its objective value $f(\boldsymbol{x})$, and updates the prior distribution with the new data point.

2.1 GPs

A GP [Rasmussen and Williams, 2006] is commonly used as the prior distribution, which regards the f value at each data point as a random variable, and assumes that all of them satisfy a joint Gaussian distribution specified by the mean value function $m(\cdot)$ and the covariance function $k(\cdot, \cdot)$. For convenience, $m(\cdot)$ is set to zero. Assume that the objective evaluation is subject to i.i.d. additive Gaussian noise, i.e., $y = f(\boldsymbol{x}) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Let $[t]$ denote the set $\{1, 2, \ldots, t\}$.

Given an observation data $D_t = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{t}$, we can obtain the posterior mean

$$\mu_t(\boldsymbol{x}) = \boldsymbol{k}_t(\boldsymbol{x})^{\mathrm{T}}(\mathbf{K}_t + \sigma^2\mathbf{I})^{-1}\boldsymbol{y}_{1:t}, \tag{1}$$

and the posterior variance

$$\sigma_t^2(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}_t(\boldsymbol{x})^{\mathrm{T}}(\mathbf{K}_t + \sigma^2\mathbf{I})^{-1}\boldsymbol{k}_t(\boldsymbol{x}), \tag{2}$$

where $\boldsymbol{k}_t(\boldsymbol{x}) = [k(\boldsymbol{x}_i, \boldsymbol{x})]_{i=1}^{t}$, $\mathbf{K}_t = [k(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j \in [t]}$ and $\boldsymbol{y}_{1:t} = [y_1; y_2; \ldots; y_t]$. For a GP, the log likelihood of observed data D_t is

$$\log \Pr(\boldsymbol{y}_{1:t} \mid \{\boldsymbol{x}_i\}_{i=1}^{t}, \boldsymbol{\theta}) = -(1/2)\boldsymbol{y}_{1:t}^{\mathrm{T}}(\mathbf{K}_t + \sigma^2\mathbf{I})^{-1}\boldsymbol{y}_{1:t} - (1/2)\log\det(\mathbf{K}_t + \sigma^2\mathbf{I}) - (t/2)\log 2\pi,$$

where $\boldsymbol{\theta}$ denote the hyper-parameters of $k(\cdot, \cdot)$, and $\det(\cdot)$ denotes the determinant of a matrix. When updating the GP model in line 5 of Algorithm 1, the hyper-parameters $\boldsymbol{\theta}$ can be updated by maximizing the log likelihood of the augmented data, or treated to be fully Bayesian.

2.2 Acquisition Functions

The data point to be evaluated in each iteration is selected by maximizing an acquisition function, which needs to trade off exploration, i.e., large posterior variances, and exploitation, i.e., large posterior means. Many acquisition functions have been proposed, and we introduce three typical ones, i.e., PI [Kushner, 1964], EI [Jones *et al.*, 1998] and UCB [Srinivas *et al.*, 2012], which will be examined in this paper.

Let $\boldsymbol{x}^+$ be the best point generated in the first $(t-1)$ iterations, and $Z = (\mu_{t-1}(\boldsymbol{x}) - f(\boldsymbol{x}^+))/\sigma_{t-1}(\boldsymbol{x})$. Let Φ and ϕ denote the cumulative distribution and probability density functions of standard Gaussian distribution, respectively. PI selects the point by maximizing the probability of improvement, i.e.,

$$\mathrm{PI}(\boldsymbol{x}) = \Pr(f(\boldsymbol{x}) > f(\boldsymbol{x}^+)) = \Phi(Z). \tag{3}$$

EI selects the data point by maximizing the expectation of improvement, i.e.,

$$\mathrm{EI}(\boldsymbol{x}) = \begin{cases} (\mu_{t-1}(\boldsymbol{x}) - f(\boldsymbol{x}^+))\Phi(Z) + \sigma_{t-1}(\boldsymbol{x})\phi(Z) \\ \qquad \text{if } \sigma_{t-1}(\boldsymbol{x}) > 0, \\ 0 \qquad \text{if } \sigma_{t-1}(\boldsymbol{x}) = 0. \end{cases} \tag{4}$$

UCB integrates the posterior mean and variance via a trade-off parameter β_t, i.e.,

$$\mathrm{UCB}(\boldsymbol{x}) = \mu_{t-1}(\boldsymbol{x}) + \beta_t^{1/2}\sigma_{t-1}(\boldsymbol{x}), \tag{5}$$

and selects the data point by maximizing this measure.

2.3 Regrets

To evaluate the performance of BO algorithms, regrets are often used. The instantaneous regret $r_t = f(\boldsymbol{x}^*) - f(\boldsymbol{x}_t)$ measures the gap of function values between an optimal solution $\boldsymbol{x}^*$ and the currently selected point $\boldsymbol{x}_t$. The simple regret $S_T = \min_{i \in [T]} r_i$ measures the gap between $\boldsymbol{x}^*$ and the best point found in the first T iterations. The cumulative regret $R_T = \sum_{i=1}^{T} r_i$ is the sum of instantaneous regrets in the first T iterations. A BO algorithm is said to be no-regret if $\lim_{T \to +\infty} R_T/T = 0$.

3 The BO-PP Framework

In BO, a GP is used to characterize the unknown objective function. The posterior variance of a GP describes the uncertainty about the unknown objective, while the posterior mean provides a closed form of the unknown objective. As the observation data augments, the posterior variance decreases and the posterior mean gets close to the unknown objective, making the GP express the unknown objective better. Thus, a straightforward way to improve the GP model is collecting more data points, which is, however, impractical, because the objective evaluation is expensive. In this section, we propose a general framework BO-PP by generating pseudo-points to improve the GP model.

As shown in Eq. (2), the posterior variance of f does not depend on the objective values, and will be decreased by adding new data points. As shown in Eq. (1), the posterior mean of f can be regarded as a linear combination of the observed objective values, and will be influenced by the error on

Algorithm 2 BO-PP Framework

Input: iteration budget T
Parameter: $\{l_i\}_{i=0}^{T-1}$, $\{\tau_i\}_{i=0}^{T-1}$
Process:
1: let $D_0 = \emptyset$ and $l_0 = 0$;
2: **for** $t = 1 : T$ **do**
3: generate l_{t-1} pseudo-points $\{(\boldsymbol{x}'_i, \hat{y}'_i)\}_{i=1}^{l_{t-1}}$;
4: re-compute $\hat{\mu}_{t-1}$ and $\hat{\sigma}_{t-1}$ by $D_{t-1} \cup \{(\boldsymbol{x}'_i, \hat{y}'_i)\}_{i=1}^{l_{t-1}}$;
5: $\boldsymbol{x}_t = \arg\max_{\boldsymbol{x}\in\mathcal{X}} acq(\boldsymbol{x})$;
6: evaluate f at $\boldsymbol{x}_t$ to obtain y_t;
7: augment the data $D_t = D_{t-1} \cup \{(\boldsymbol{x}_t, y_t)\}$ and update the GP model
8: **end for**

where each pseudo-point in the t-th iteration has distance τ_{t-1} to some observed data point in D_{t-1}, and takes the same objective value as the observed one.

the objective values of new data points. Inspired by the Lipschitz assumption, i.e., close data points have close objective values, the pseudo-points are selected to be neighbors of the observed data points, and take the same objective values as the observed ones.

The BO-PP framework is described in Algorithm 2. Before selecting the next data point in line 5, BO-PP generates a few pseudo-points to re-compute the posterior mean and variance of the GP model in lines 3-4, rather than directly using the GP model updated in the last iteration. After evaluating a new data point in line 6, the hyper-parameters of the covariance function employed by the GP model will be updated in line 7 using the truly observed data points by far. Note that the pseudo-points are only used to re-compute the posterior mean and variance.

The way of generating pseudo-points can be diverse, e.g., randomly sampling a point with distance τ from some observed data point.* The only requirement is that the pseudo-point takes the same objective value as the corresponding observed data point, which does not increase the evaluation cost. The number l_t of pseudo-points and the distance τ_t employed in each iteration could affect the performance of the algorithm. For example, as τ_t decreases, the error on the objective values of pseudo-points will decrease, whereas the reduction on the posterior variance will also decrease. Their relationship will be analyzed in theoretical analysis, and we will provide an effective way of setting l_t and τ_t in experiments. Note that BO-PP can be equipped with any acquisition function.

4 Theoretical Analysis

In this section, we theoretically analyze the performance of BO-PP w.r.t. the acquisition function UCB, called UCB-PP. Specifically, we prove that the cumulative regret R_T of UCB-PP can be generally upper bounded.

We first give some notations that will be used in the following analysis. Let μ_t and σ_t denote the posterior mean and variance after obtaining D_t; let $\hat{\mu}_t$ and $\hat{\sigma}_t$ denote the posterior mean and variance after adding pseudo-points $\{(\boldsymbol{x}'_i, \hat{y}'_i)\}_{i=1}^{l_t}$ into D_t; let $\tilde{\mu}_t$ and $\tilde{\sigma}_t$ denote the posterior mean and variance after adding pseudo-points with true observed objective values, i.e., $\{(\boldsymbol{x}'_i, y'_i))\}_{i=1}^{l_t}$, where $y'_i = f(\boldsymbol{x}'_i)+\epsilon'_i$ with $\epsilon'_i \sim \mathcal{N}(0, \sigma^2)$. Some notations about pseudo-points: $\hat{\boldsymbol{y}}'_{1:l_t} = [\hat{y}'_1; \hat{y}'_2; \ldots; \hat{y}'_{l_t}]$; $\boldsymbol{y}'_{1:l_t} = [y'_1; y'_2; \ldots; y'_{l_t}]$; $\boldsymbol{k}'_{l_t}(\boldsymbol{x}) = [k(\boldsymbol{x}'_i, \boldsymbol{x})]_{i=1}^{l_t}$; $\mathbf{K}'_{l_t} = [k(\boldsymbol{x}'_i, \boldsymbol{x}'_j)]_{i,j\in[l_t]}$; $\widetilde{\mathbf{K}}_{t,l_t} = [k(\boldsymbol{x}_i, \boldsymbol{x}'_j)]_{i\in[t],j\in[l_t]}$; $\boldsymbol{p}(\boldsymbol{x}) = \widetilde{\mathbf{K}}_{t,l_t}^{\mathrm{T}}(\mathbf{K}_t + \sigma^2\mathbf{I})^{-1}\boldsymbol{k}_t(\boldsymbol{x}) - \boldsymbol{k}'_{l_t}(\boldsymbol{x})$; $\mathbf{M} = (\mathbf{K}'_{l_t} - \widetilde{\mathbf{K}}_{t,l_t}^{\mathrm{T}}(\mathbf{K}_t + \sigma^2\mathbf{I})^{-1}\widetilde{\mathbf{K}}_{t,l_t} + \sigma^2\mathbf{I})^{-1}$. For convenience of analysis, assume $k(\boldsymbol{x}, \boldsymbol{x}) = 1$.

Let A be a finite subset of $\mathcal{X}$, $\boldsymbol{f}_A$ denote their true objective values (which are actually random variables satisfying the posterior Gaussian distribution over the true objective values), and $\boldsymbol{y}_A$ denote the noisy observations. Let PP denote all generated pseudo-points, and $\hat{\boldsymbol{y}}_{PP}$ denote their selected objective values. Note that $\hat{\boldsymbol{y}}_{PP}$ are random variables, as they are actually the noisy observations of the objective values of PP's neighbor observed points. Let $\gamma'_T = \max_{A:|A|=T} I(\boldsymbol{y}_A; \boldsymbol{f}_A) - \min_{A:|A|=T,PP} I(\boldsymbol{y}_A; \hat{\boldsymbol{y}}_{PP})$, where $I(\cdot;\cdot)$ denotes the mutual information.

Theorem 1 gives an upper bound of UCB-PP on the cumulative regret R_T. As the analysis of UCB in [Srinivas *et al.*, 2012], Assumption 1 is required, implying

$$\Pr(\forall \boldsymbol{x}, \boldsymbol{x}' : |f(\boldsymbol{x}) - f(\boldsymbol{x}')| \le L\|\boldsymbol{x} - \boldsymbol{x}'\|_1) \ge 1 - dae^{-(L/b)^2}. \quad (6)$$

Assumption 1. *Suppose the kernel $k(\cdot,\cdot)$ satisfies the following probability bound on the derivatives of f: for some constants $a, b > 0$, $\forall j \in [d] : \Pr(sup_{\boldsymbol{x}\in\mathcal{X}}|\partial f/\partial x_j| > L) \le ae^{-(L/b)^2}$.*

Theorem 1. *Let $\mathcal{X} \subset [0, r]^d$, $\delta \in (0, 1)$, and set β_t in Eq. (5) as $\beta_t = 2\log(2\pi^2 t^2/(3\delta)) + 2d\log(t^2 dbr\sqrt{\log(4da/\delta)})$. Running UCB-PP for T iterations, it holds that*

$$\Pr\Big(R_T \le \sqrt{CT\beta_T\gamma'_T} + 2 + 2\textstyle\sum_{t=1}^{T}\Delta_m(l_{t-1}, \tau_{t-1})\Big) \ge 1 - \delta, \quad (7)$$

where $C = 8/\log(1 + \sigma^{-2})$, and $\Delta_m(l_t, \tau_t) = l_t^2\sqrt{1+\sigma^{-2}}\Big(bd\tau_t\sqrt{\log(4da/\delta)}/\sigma + 2\sqrt{\log\frac{4\sum_{t=0}^{T-1} l_t}{\delta}}\Big)$.

Lemma 1 bounds the error on the posterior mean led by the incorrect objective values of pseudo-points, which will be used in the proof of Theorem 1. Its proof is provided in the supplementary material due to space limitation.

Lemma 1. *After obtaining D_t in UCB-PP, the difference on the posterior mean by adding pseudo-points, i.e., $\{(\boldsymbol{x}'_i, \hat{y}'_i)\}_{i=1}^{l_t}$, and that with true observed objective values, i.e., $\{(\boldsymbol{x}'_i, y'_i))\}_{i=1}^{l_t}$, is $\hat{\mu}_t(\boldsymbol{x}) - \tilde{\mu}_t(\boldsymbol{x}) = -\boldsymbol{p}(\boldsymbol{x})^{\mathrm{T}}\mathbf{M}(\hat{\boldsymbol{y}}'_{1:l_t} - \boldsymbol{y}'_{1:l_t})$. Furthermore, it holds that*

$$\Pr\big(\forall 0 \le t \le T-1, \forall \boldsymbol{x} \in \mathcal{X} : |\hat{\mu}_t(\boldsymbol{x}) - \tilde{\mu}_t(\boldsymbol{x})| \le \Delta_m(L, l_t, \tau_t)\big) \ge 1 - dae^{-(L/b)^2} - \delta/4,$$

where $\Delta_m(L, l_t, \tau_t) = l_t^2\sqrt{1+\sigma^{-2}}\Big(\frac{Ld\tau_t}{\sigma} + 2\sqrt{\log\frac{4\sum_{t=0}^{T-1} l_t}{\delta}}\Big)$.

*Here, two data points $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d$ have distance τ means that $\forall i \in [d] : |x_i - x'_i| = \tau$.

The proof of Theorem 1 is inspired by that of Theorem 2 in [Srinivas *et al.*, 2012], which gives an upper bound of UCB on the cumulative regret R_T. Their proof intuition is mainly that the instantaneous regret r_t can be upper bounded by the width of confidence interval of $f(\boldsymbol{x}_t)$, relating to the posterior variance. The generation of pseudo-points will introduce another quantity into the upper bound on r_t, characterized by the error on the posterior mean in Lemma 1.

Proof of Theorem 1. According to Assumption 1 and $\beta_t = 2\log(2\pi^2 t^2 (dt^2 rL)^d/(3\delta))$, where $L = b\sqrt{\log(4da/\delta)}$, we can apply Lemma 5.7 in [Srinivas *et al.*, 2012] to derive that

$$\Pr\big(\forall t \geq 1 : |f(\boldsymbol{x}^*) - \tilde{\mu}_{t-1}([\boldsymbol{x}^*]_t)| \leq \beta_t^{1/2}\tilde{\sigma}_{t-1}([\boldsymbol{x}^*]_t) + 1/t^2\big) \geq 1 - \delta/2, \tag{8}$$

where $[\boldsymbol{x}^*]_t$ denotes the discretized data point closest to $\boldsymbol{x}^*$ in the t-th iteration. Note that $\Delta_m(l_t, \tau_t)$ is just $\Delta_m(L, l_t, \tau_t)$ with $L = b\sqrt{\log(4da/\delta)}$ in Lemma 1. By the definition of r_t, we have, $\forall t \geq 1$:

$$\begin{aligned}
r_t &= f(\boldsymbol{x}^*) - f(\boldsymbol{x}_t) \\
&\leq \beta_t^{1/2}\tilde{\sigma}_{t-1}([\boldsymbol{x}^*]_t) + \tilde{\mu}_{t-1}([\boldsymbol{x}^*]_t) - f(\boldsymbol{x}_t) + 1/t^2 \\
&\leq \beta_t^{1/2}\tilde{\sigma}_{t-1}([\boldsymbol{x}^*]_t) + \hat{\mu}_{t-1}([\boldsymbol{x}^*]_t) - f(\boldsymbol{x}_t) + 1/t^2 \\
&\quad + \Delta_m(l_{t-1}, \tau_{t-1}) \\
&= \beta_t^{1/2}\hat{\sigma}_{t-1}([\boldsymbol{x}^*]_t) + \hat{\mu}_{t-1}([\boldsymbol{x}^*]_t) - f(\boldsymbol{x}_t) + 1/t^2 \\
&\quad + \Delta_m(l_{t-1}, \tau_{t-1}) \\
&\leq \beta_t^{1/2}\hat{\sigma}_{t-1}(\boldsymbol{x}_t) + \hat{\mu}_{t-1}(\boldsymbol{x}_t) - f(\boldsymbol{x}_t) + 1/t^2 \\
&\quad + \Delta_m(l_{t-1}, \tau_{t-1}) \\
&\leq \beta_t^{1/2}\hat{\sigma}_{t-1}(\boldsymbol{x}_t) + \tilde{\mu}_{t-1}(\boldsymbol{x}_t) - f(\boldsymbol{x}_t) + 1/t^2 \\
&\quad + 2\Delta_m(l_{t-1}, \tau_{t-1}) \\
&\leq 2\beta_t^{1/2}\hat{\sigma}_{t-1}(\boldsymbol{x}_t) + 1/t^2 + 2\Delta_m(l_{t-1}, \tau_{t-1}),
\end{aligned}$$

where the first inequality holds with probability at least $1 - \delta/2$ by Eq. (8), the second and fourth inequalities hold with probability at least $1 - dae^{-(L/b)^2} - \delta/4 = 1 - \delta/2$ by Lemma 1, the equality holds because the posterior variance in Eq. (2) does not depend on the objective values, leading to $\forall \boldsymbol{x} : \hat{\sigma}_{t-1}(\boldsymbol{x}) = \tilde{\sigma}_{t-1}(\boldsymbol{x})$, the third inequality holds because $\boldsymbol{x}_t$ is selected by maximizing $\hat{\mu}_{t-1}(\boldsymbol{x}) + \beta_t^{1/2}\hat{\sigma}_{t-1}(\boldsymbol{x})$ in Eq. (5), and the last inequality holds with probability at least $1 - \delta/4$ by Lemma 5.5 in [Srinivas *et al.*, 2012]. Note that to prove Lemma 5.7 in [Srinivas *et al.*, 2012] and Lemma 1, Assumption 1, i.e., Eq. (6), is both used; thus, the probability $dae^{-(L/b)^2} = \delta/4$ has been repeated. By the union bound, we have

$$\begin{aligned}
&\Pr\big(\forall t \geq 1 : r_t \leq 2\beta_t^{1/2}\hat{\sigma}_{t-1}(\boldsymbol{x}_t) + 1/t^2 + 2\Delta_m(l_{t-1}, \tau_{t-1})\big) \\
&\geq 1 - \delta/2 - \delta/4 - \delta/4 = 1 - \delta,
\end{aligned}$$

implying

$$\Pr\Big(R_T = \textstyle\sum_{t=1}^T r_t \leq \sum_{t=1}^T \big(2\beta_t^{1/2}\hat{\sigma}_{t-1}(\boldsymbol{x}_t) + 1/t^2 + 2\Delta_m(l_{t-1}, \tau_{t-1})\big)\Big) \geq 1 - \delta.$$

By the Cauchy–Schwarz inequality, $C = 8/\log(1 + \sigma^{-2})$ and $\forall t \leq T : \beta_t \leq \beta_T$, we have

$$\begin{aligned}
\textstyle\sum_{t=1}^T 2\beta_t^{1/2}\hat{\sigma}_{t-1}(\boldsymbol{x}_t) &\leq \sqrt{T\textstyle\sum_{t=1}^T 4\beta_t\hat{\sigma}_{t-1}^2(\boldsymbol{x}_t)} \\
&\leq \sqrt{\frac{CT\beta_T}{2}\textstyle\sum_{t=1}^T \log(1 + \sigma^{-2}\hat{\sigma}_{t-1}^2(\boldsymbol{x}_t))}.
\end{aligned}$$

Let PP_t denote the pseudo-points generated in the t-th iteration, and $\hat{\boldsymbol{y}}_{PP_t}$ denote their selected objective values. Let $H(\cdot)$ denote the entropy. We have

$$\begin{aligned}
&\frac{1}{2}\textstyle\sum_{t=1}^T \log(1 + \sigma^{-2}\hat{\sigma}_{t-1}^2(\boldsymbol{x}_t)) + H(\boldsymbol{y}_{1:T}|\boldsymbol{f}_{1:T}) \\
&= \frac{1}{2}\textstyle\sum_{t=1}^T \log(1 + \sigma^{-2}\hat{\sigma}_{t-1}^2(\boldsymbol{x}_t)) + \frac{1}{2}\log(\det(2\pi e\sigma^2\mathbf{I})) \\
&= \frac{1}{2}\textstyle\sum_{t=1}^T \log(2\pi e(\sigma^2 + \hat{\sigma}_{t-1}^2(\boldsymbol{x}_t))) \\
&= H(y_1 \mid \hat{\boldsymbol{y}}_{PP_1}) + H(y_2 \mid y_1, \hat{\boldsymbol{y}}_{PP_2}) + \cdots \\
&\quad + H(y_T \mid \boldsymbol{y}_{1:T-1}, \hat{\boldsymbol{y}}_{PP_T}) \\
&= H(y_1 \mid \hat{\boldsymbol{y}}_{PP}) + H(y_2 \mid y_1, \hat{\boldsymbol{y}}_{PP}) + \cdots \\
&\quad + H(y_T \mid \boldsymbol{y}_{1:T-1}, \hat{\boldsymbol{y}}_{PP}) \\
&= H(\boldsymbol{y}_{1:T} \mid \hat{\boldsymbol{y}}_{PP}),
\end{aligned}$$

where the first equality holds because $f(\boldsymbol{x})$ is subject to additive Gaussian noise $\mathcal{N}(0, \sigma^2)$. Thus,

$$\begin{aligned}
&\frac{1}{2}\textstyle\sum_{t=1}^T \log(1 + \sigma^{-2}\hat{\sigma}_{t-1}^2(\boldsymbol{x}_t)) \\
&= H(\boldsymbol{y}_{1:T} \mid \hat{\boldsymbol{y}}_{PP}) - H(\boldsymbol{y}_{1:T} \mid \boldsymbol{f}_{1:T}) \\
&= H(\hat{\boldsymbol{y}}_{PP} \mid \boldsymbol{y}_{1:T}) - H(\hat{\boldsymbol{y}}_{PP}) + H(\boldsymbol{y}_{1:T}) - H(\boldsymbol{y}_{1:T} \mid \boldsymbol{f}_{1:T}) \\
&= I(\boldsymbol{y}_{1:T}; \boldsymbol{f}_{1:T}) - I(\boldsymbol{y}_{1:T}; \hat{\boldsymbol{y}}_{PP}) \leq \gamma'_T.
\end{aligned}$$

Considering $\sum_{t\geq 1} 1/t^2 = \pi^2/6 < 2$, Eq. (7) holds. Thus, the theorem holds. □

Adding pseudo-points is to improve the GP model when the number of observed data points is not large. After UCB-PP runs many iterations, there are already enough observed points, and thus pseudo-points are not needed. That is, UCB-PP will only add pseudo-points in a finite number of iterations, denoted by T_0. This implies that $\forall t \geq T_0$, $l_t = 0$, leading to $\Delta_m(l_t, \tau_t) = 0$. Thus, $\lim_{T\to+\infty} R_T/T = 0$, implying that UCB-PP is no-regret.

Under the same assumption, it has been proved [Srinivas *et al.*, 2012] that the cumulative regret R_T of UCB satisfies

$$\Pr\big(R_T \leq \sqrt{CT\beta_T\gamma_T} + 2\big) \geq 1 - \delta, \tag{9}$$

where $\gamma_T = \max_{A:|A|=T} I(\boldsymbol{y}_A; \boldsymbol{f}_A)$, and the other parameters have the same meaning as that in Theorem 1. Without generating pseudo-points, $\forall t \geq 0 : l_t = 0 \wedge I(\boldsymbol{y}_A; \hat{\boldsymbol{y}}_{PP}) = 0$, and thus, $\gamma'_T = \gamma_T \wedge \Delta_m(l_t, \tau_t) = 0$, implying that Eq. (7) specializes to Eq. (9). Thus, we have:

Remark 1. *Our bound on R_T of UCB-PP is a generalization of the bound on R_T of UCB in [Srinivas* et al.*, 2012].*

As $\gamma'_T \leq \gamma_T$, the comparison between Eqs. (7) and (9) suggests that the generation of pseudo-points can be helpful if the negative influence of introducing the error on the posterior mean, i.e., introducing the term $\Delta_m(l_t, \tau_t)$, can be compensated by the positive influence of reducing the posterior variance, i.e., introducing the term $I(\boldsymbol{y}_{1:T}; \hat{\boldsymbol{y}}_{PP})$.

	Function	UCB	UCB-PP01	UCB-PP001	UCB-PP0001
S_T	*Dropwave*	0.2710±0.1311	**0.2232±0.1053**	**0.1630±0.1014**	**0.2121±0.1038**
	Griewank	0.2357±0.2125	**0.2272±0.1644**	**0.2350±0.1690**	**0.2085±0.1177**
	Hart6	1.0256±0.3498	1.0565±0.3620	1.0868±0.3153	**0.9276±0.3307**
	Rastrigin	3.3492±3.2602	3.6975±2.7991	3.5124±2.4124	**3.0077±2.3245**
f	*SVM_wine*	0.6182±0.0029	**0.6186±0.0030**	**0.6186±0.0036**	**0.6189±0.0042**
	NN_wine	0.9149±0.0004	**0.9151±0.0005**	**0.9151±0.0004**	**0.9151±0.0004**
	NN_cancer	0.9585±0.0006	**0.9589±0.0006**	**0.9589±0.0006**	**0.9590±0.0006**
	NN_housing	8.6733±1.6916	**8.6776±1.7149**	**9.0691±1.8656**	**8.7216±1.8076**
	Function	PI	PI-PP01	PI-PP001	PI-PP0001
S_T	*Dropwave*	0.1526±0.1534	**0.1221±0.1462**	**0.1251±0.1355**	**0.1457±0.1539**
	Griewank	0±0	**0±0**	**0±0**	**0±0**
	Hart6	0.5795±0.2959	**0.4558±0.1048**	**0.5599±0.0982**	**0.5500±0.2529**
	Rastrigin	0.0524±0.2285	**0.0524±0.2285**	**0±0**	**0.0524±0.2285**
f	*SVM_wine*	0.6192±0.0037	0.6176±0.0025	**0.6204±0.0050**	**0.6208±0.0053**
	NN_wine	0.9140±0.0011	**0.9143±0.0006**	**0.9140±0.0008**	0.9138±0.0008
	NN_cancer	0.9571±0.0024	**0.9578±0.0020**	**0.9576±0.0024**	**0.9574±0.0024**
	NN_housing	7.5570±1.4822	**7.9702±1.4000**	**7.8585±1.5515**	**7.6147±1.3592**
	Function	EI	EI-PP01	EI-PP001	EI-PP0001
S_T	*Dropwave*	0.2557±0.1720	**0.1924±0.0818**	**0.2307±0.1461**	**0.2276±0.1752**
	Griewank	0.3098±0.1722	**0.3028±0.1005**	0.3187±0.1594	**0.2729±0.1471**
	Hart6	0.6652±0.2685	**0.6050±0.2328**	**0.6028±0.1656**	0.6828±0.3081
	Rastrigin	3.3069±2.4955	**2.6602±2.2063**	**3.0492±1.5602**	**3.1987±2.3818**
f	*SVM_wine*	0.6189±0.0037	0.6182±0.0035	**0.6198±0.0037**	0.6179±0.0034
	NN_wine	0.9149±0.0006	**0.9150±0.0005**	**0.9150±0.0004**	0.9148±0.0005
	NN_cancer	0.9587±0.0006	**0.9589±0.0006**	**0.9588±0.0006**	**0.9587±0.0006**
	NN_housing	8.1780±1.8382	8.0277±1.3844	8.0471±1.5024	**8.1992±1.7971**

Table 1: The results (mean±std.) of BO-PP and BO on synthetic benchmark functions and real-world optimization problems, when reaching the iteration budget. S_T: the smaller, the better; f: the larger, the better. The bolded values denote that BO-PP is no worse than BO. UCB, PI and EI are tested.

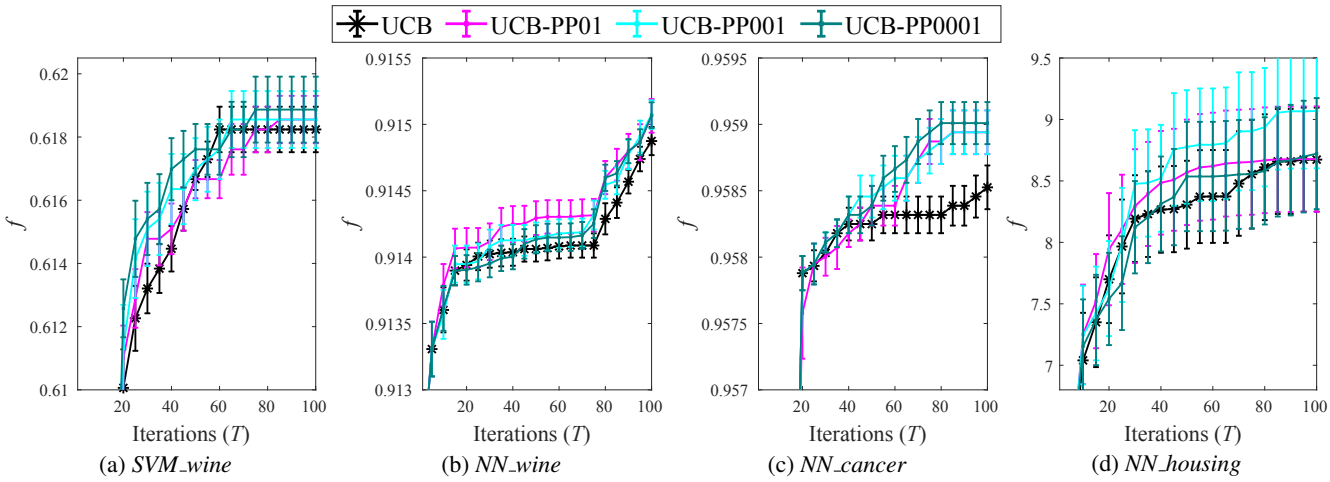
5 Empirical Study

In this section, we empirically compare BO-PP with BO. Three common acquisition functions, i.e., UCB, PI and EI, are used. The ARD squared exponential kernel is employed, whose hyper-parameters are tuned by maximum likelihood estimation (MLE), and the acquisition function is maximized via the DIRECT algorithm [Jones *et al.*, 1993]. To alleviate the "cold start" issue, each algorithm starts with five random initial points. To compare BO-PP with BO on each problem, we repeat their running 20 times independently and report the average results; in each running, BO-PP and BO use the same five random initial points. The noise level is set to $\sigma^2 = 0.0001$, and the iteration budget is set to 100.

In the $(t+1)$-th iteration of BO-PP, for each point in D_t, one pseudo-point is generated by randomly sampling within its distance τ_t and taking the same function value; thus, $l_t = |D_t|$. To control the error of objective values with pseudo-points increasing, τ_t is set to $r\tau_0/(dl_t)$, which decreases with l_t. Note that r corresponds to the width of each dimension of the search domain. τ_0 is set to a small value. We will use 0.01, 0.001 and 0.0001 to explore its influence, and the corresponding algorithms are denoted as BO-PP01, BO-PP001 and BO-PP0001, respectively.

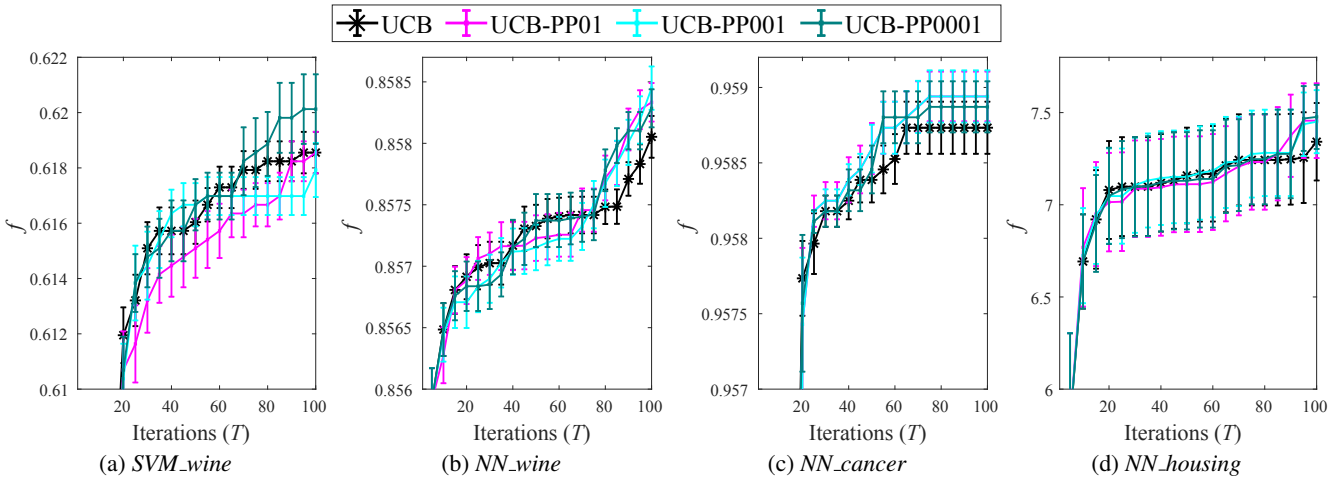
We use four common synthetic benchmark functions: *Dropwave*, *Griewank*, *Hart6* and *Rastrigin*, whose dimensions are 2, 2, 6 and 2, respectively. Their search domains are scaled to $[-1, 1]^d$. As the minima are known, the simple regret S_T is used as the metric. We also employ four real-world optimization problems, widely used in BO experiments [Springenberg *et al.*, 2016; Wang and Jegelka, 2017; Ru *et al.*, 2018]. The first is to tune the hyper-parameters, i.e., box constraint $C \in [0.001, 1000]$ and kernel scale $l \in [0.0001, 1]$, of SVM for classification on the data set *Wine quality* (1,599 #inst, 11 #feat). The second is to tune the hyper-parameters of 1-hidden-layer neural network (NN) for this task. The NN is trained by backpropagation, and the hyper-parameters are the number of neurons $n \in [1, 100]$ and the learning rate $l_r \in [0.000001, 1]$. The last two problems are to tune the hyper-parameters of 1-hidden-layer NN for classification on *Breast cancer* (699 #inst, 9 #feat) and regression on *Boston housing* (506 #inst, 13 #feat), respectively. The NN is trained by Levenberg-Marquardt optimization, and there are four hyper-parameters: $n \in [1, 100]$, the damping factor $\mu \in [0.000001, 100]$, the μ-decrease and μ-increase factors $\mu_{dec} \in [0.01, 1]$, $\mu_{inc} \in [1.01, 20]$. All data sets are randomly split into training/validation/test sets with ratio 0.7/0.2/0.1, and the performance on validation sets is used as the objective f. For classification, f is the classification accuracy; for regression, f equals 20 minus the regression L2-loss.

For UCB, β_t in Eq. (5) is set to $2\log(t^{d/2+2}\pi^2/3\delta)$ where $\delta = 0.1$, as suggested in [Brochu *et al.*, 2010; Srinivas *et al.*, 2012]. For PI and EI, the best observed function value by far is used as $f(\boldsymbol{x}^+)$ in Eqs. (3) and (4). The results are summarized in Table 1. We can observe that UCB-PP0001



(a) *SVM_wine* (b) *NN_wine* (c) *NN_cancer* (d) *NN_housing*

Figure 1: The results (mean±(1/4)std.) of UCB-PP and UCB on real-world optimization problems. f: the larger, the better.



(a) *SVM_wine* (b) *NN_wine* (c) *NN_cancer* (d) *NN_housing*

Figure 2: The results (mean±(1/4)std.) of UCB-PP and UCB with the Gaussian kernel on real-world optimization problems. f: the larger, the better.

is always better than UCB, and UCB-PP01/UCB-PP001 surpasses UCB in most cases, disclosing that the performance of UCB-PP is not very sensitive to the distance τ_t. Also, PI-PP and EI-PP perform better than PI and EI, respectively, in most cases, showing the applicability of generating pseudo-points.

Furthermore, we plot the curves of the simple regret S_T or the objective f over iterations for each algorithm on each problem. Figure 1 shows the curves of UCB-PP and UCB on real-world problems. It can be observed that on each problem, there is at least one curve of UCB-PP almost always above that of UCB, implying that UCB-PP can consistently outperform UCB during the running process. The other five figures, showing similar observations, are provided in the supplementary material due to space limitation.

To examine the robustness of BO-PP against kernels, we use the Gaussian kernel with hyper-parameters tuned by MLE. We compare UCB-PP with UCB on real-world optimization problems. Figure 2 shows that UCB-PP can be better than UCB except UCB-PP01/UCB-PP001 on *SVM_wine*.

6 Conclusion

In this paper, we propose a general framework BO-PP by generating pseudo-points to improve the GP model of BO. BO-PP can be implemented with any acquisition function. Equipped with UCB, we prove that the cumulative regret of BO-PP can be well bounded. This bound generalizes the well-known bound of UCB. Experiments with UCB, PI and EI on synthetic as well as real-world optimization problems show the superior performance of BO-PP over BO. It is expected that the generation of pseudo-points can be helpful for more BO algorithms. Note that the dimensionality of problems tested in our experiments is low. Thus, studying the effectiveness of BO-PP on high-dimensional optimization problems is an interesting topic.

It is also noted that the added pseudo-points take the same function values with their neighbor observed data points, requiring the function to vary smoothly locally. Thus, it is interesting to study strategies of improving BO when the function can fluctuate widely in the future.

References

[Azimi *et al.*, 2010] J. Azimi, A. Fern, and X. Z. Fern. Batch Bayesian optimization via simulation matching. In *Advances in Neural Information Processing Systems 23 (NIPS'10)*, pages 109–117, Vancouver, Canada, 2010.

[Brochu *et al.*, 2010] E. Brochu, V. M. Cora, and N. de Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *CoRR abs/1012.2599*, 2010.

[Denil *et al.*, 2012] M. Denil, L. Bazzani, H. Larochelle, and N. de Freitas. Learning where to attend with deep architectures for image tracking. *Neural Computation*, 24(8):2151–2184, 2012.

[Desautels *et al.*, 2014] T. Desautels, A. Krause, and J. W. Burdick. Parallelizing exploration-exploitation tradeoffs in Gaussian process bandit optimization. *Journal of Machine Learning Research*, 15:3873–3923, 2014.

[Garnett *et al.*, 2010] R. Garnett, M. A. Osborne, and S. J. Roberts. Bayesian optimization for sensor set selection. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN'10)*, pages 209–219, Stockholm, Sweden, 2010.

[González *et al.*, 2016] J. González, Z. Dai, P. Hennig, and N. D. Lawrence. Batch Bayesian optimization via local penalization. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS'16)*, pages 648–657, Cadiz, Spain, 2016.

[Hennig and Schuler, 2012] P. Hennig and C. J. Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13:1809–1837, 2012.

[Hernández-Lobato *et al.*, 2014] J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems 27 (NIPS'14)*, pages 918–926, Montreal, Canada, 2014.

[Jones *et al.*, 1993] D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.

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

[Kandasamy *et al.*, 2015] K. Kandasamy, J. G. Schneider, and B. Póczos. High dimensional Bayesian optimisation and bandits via additive models. In *Proceedings of the 32nd International Conference on Machine Learning (ICML'15)*, pages 295–304, Lille, France, 2015.

[Kushner, 1964] H. J. Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1):97–106, 1964.

[Lyu *et al.*, 2018] W. Lyu, F. Yang, C. Yan, D. Zhou, and X. Zeng. Batch Bayesian optimization via multi-objective acquisition ensemble for automated analog circuit design. In *Proceedings of the 35th International Conference on Machine Learning (ICML'18)*, pages 3306–3314, Stockholm, Sweden, 2018.

[Martinez-Cantin *et al.*, 2007] R. Martinez-Cantin, N. de Freitas, A. Doucet, and J. A. Castellanos. Active policy learning for robot planning and exploration under uncertainty. In *Robotics: Science and Systems III (RSS'07)*, pages 321–328, Atlanta, GA, 2007.

[Mockus, 1994] J. Mockus. Application of Bayesian approach to numerical methods of global and stochastic optimization. *Journal of Global Optimization*, 4(4):347–365, 1994.

[Mutny and Krause, 2018] M. Mutny and A. Krause. Efficient high dimensional Bayesian optimization with additivity and quadrature fourier features. In *Advances in Neural Information Processing Systems 31 (NeurIPS'18)*, pages 9005–9016, Montreal, Canada, 2018.

[Rasmussen and Williams, 2006] C. E. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, Cambridge, MA, 2006.

[Ru *et al.*, 2018] B. X. Ru, M. McLeod, D. Granziol, and M. A. Osborne. Fast information-theoretic Bayesian optimisation. In *Proceedings of the 35th International Conference on Machine Learning (ICML'18)*, pages 4381–4389, Stockholm, Sweden, 2018.

[Shah and Ghahramani, 2015] A. Shah and Z. Ghahramani. Parallel predictive entropy search for batch global optimization of expensive objective functions. In *Advances in Neural Information Processing Systems 28 (NIPS'15)*, pages 3330–3338, Montreal, Canada, 2015.

[Snoek *et al.*, 2012] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25 (NIPS'12)*, pages 2951–2959, Lake Tahoe, NV, 2012.

[Springenberg *et al.*, 2016] J. T. Springenberg, A. Klein, S. Falkner, and F. Hutter. Bayesian optimization with robust Bayesian neural networks. In *Advances in Neural Information Processing Systems 29 (NIPS'16)*, pages 4134–4142, Barcelona, Spain, 2016.

[Srinivas *et al.*, 2012] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger. Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

[Wang and Jegelka, 2017] Z. Wang and S. Jegelka. Max-value entropy search for efficient Bayesian optimization. In *Proceedings of the 34th International Conference on Machine Learning (ICML'17)*, pages 3627–3635, Sydney, Australia, 2017.

[Wang *et al.*, 2013] Z. Wang, M. Zoghi, F. Hutter, D. Matheson, and N. de Freitas. Bayesian optimization in high dimensions via random embeddings. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13)*, pages 1778–1784, Beijing, China, 2013.

[Wang *et al.*, 2016] Z. Wang, B. Zhou, and S. Jegelka. Optimization as estimation with Gaussian processes in bandit settings. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS'16)*, pages 1022–1031, Cadiz, Spain, 2016.

[Wang *et al.*, 2017] Z. Wang, C. Li, S. Jegelka, and P. Kohli. Batched high-dimensional Bayesian optimization via structural kernel learning. In *Proceedings of the 34th International Conference on Machine Learning (ICML'17)*, pages 3656–3664, Sydney, Australia, 2017.