# Supplemental Document

## Sequential Line Search for Efficient Visual Design Optimization by Crowds

YUKI KOYAMA, ISSEI SATO, DAISUKE SAKAMOTO, and TAKEO IGARASHI, The University of Tokyo

This supplemental document provides the details of equations and our implementation for reproducibility. Also, we include additional discussions and figures that are useful for better understanding of our method.

## 1 BAYESIAN OPTIMIZATION: FUNDAMENTALS AND OUR IMPLEMENTATION

We have briefly introduced an overview of standard Bayesian optimization techniques in our main paper. Here, we provide more details of them. Note that readers can also find general and comprehensive introductions in [Brochu et al. 2010b; Shahriari et al. 2016].

### 1.1 Overview

Suppose that $\mathcal{A}$ is a $d$-dimensional bounded space, $f : \mathcal{A} \to \mathbb{R}$ is an unknown black-box function, and we want to find its maximum:

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathcal{A}}{\arg\max}\ f(\mathbf{x}). \tag{1}$$

Suppose as well that the function value $f(\mathbf{x})$ can be computed for an arbitrary point $\mathbf{x}$, but $f(\cdot)$ is an *expensive-to-evaluate* function, *i.e.,* it entails a significant computational cost to evaluate the function value. Thus, while there are many optimization algorithms that can be used for solving this maximization problem (*e.g.,* the DIRECT algorithm [Jones et al. 1993]), here we are especially interested in making the number of necessary function evaluations as small as possible.

Suppose that we currently have a set of $t$ function-value observations:

$$\mathcal{D}_t = \{(\mathbf{x}_i, f_i)\}_{i=1}^{t}, \tag{2}$$

where $f_i = f(\mathbf{x}_i)$. Intuitively, for each iteration in Bayesian optimization, the next evaluation point $\mathbf{x}_{t+1}$ is determined such that it is "the one most worth observing" based on the previous data $\mathcal{D}_t$. Suppose that $a : \mathcal{A} \to \mathbb{R}$ is a function that quantifies the "worthiness" of the next sampling candidate. We call this function an *acquisition function*. For each iteration, the system computes the maximization of the acquisition function to determine the most effective next sampling point:

$$\mathbf{x}_{t+1} = \underset{\mathbf{x}\in\mathcal{A}}{\arg\max}\ a(\mathbf{x}; \mathcal{D}_t). \tag{3}$$

The following subsections explain how to model and calculate such an acquisition function. Before introducing the detailed equations of the acquisition function, we begin with a prior assumption put on the objective function, based on which the acquisition function is calculated.

### 1.2 Gaussian Process Prior

In Bayesian optimization, the *Gaussian process* (GP) prior is often assumed on $f(\cdot)$. According to [Ebden 2015], a GP is described as follows:

> *"Formally, a Gaussian process generates data located throughout some domain such that any finite subset of the range follows a multivariate Gaussian distribution."*

This is expressed as

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \tag{4}$$

where $m : \mathcal{A} \to \mathbb{R}$ is the *mean function* and $k : \mathcal{A} \times \mathcal{A} \to \mathbb{R}^+$ is the *covariance function* of the GP. When prior knowledges about $f(\cdot)$ are available, $m(\cdot)$ can be set to reflect those knowledges (*e.g.,* [Brochu et al. 2010a]). In this paper, as we do not assume any domain-specific prior knowledge, we simply set

$$m(\mathbf{x}) = 0. \tag{5}$$

For the covariance function representation, we use the *automatic relevance determination* (ARD) *squared exponential kernel* [Rasmussen and Williams 2006]:

$$k(\mathbf{x}, \mathbf{x}') = \theta_{d+1} \exp\left\{-\frac{1}{2}\sum_{i=1}^{d}\frac{(x_i - x_i')^2}{\theta_i^2}\right\} + \theta_{d+2}\delta(\mathbf{x}, \mathbf{x}'), \tag{6}$$

where $\boldsymbol{\theta} = \{\theta_i\}_{i=1}^{d+2}$ are the model hyperparameters that should be determined somehow, which will be discussed later, and $\delta(\cdot,\cdot)$ is the Kronecker-Delta function.

Since any data should follow a multivariate Gaussian distribution under the GP prior, an unobserved function value $f(\mathbf{x}_*)$ on an arbitrary parameter set $\mathbf{x}_*$ is considered to follow the distribution:

$$\begin{bmatrix}\mathbf{f}\\ f(\mathbf{x}_*)\end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix}\mathbf{K} & \mathbf{k}\\ \mathbf{k}^T & k(\mathbf{x}_*, \mathbf{x}_*)\end{bmatrix}\right), \tag{7}$$

where

$$\mathbf{f} = \begin{bmatrix} f_1 & \cdots & f_N\end{bmatrix}^T, \tag{8}$$

$$\mathbf{k} = \begin{bmatrix} k(\mathbf{x}_*, \mathbf{x}_1) & \cdots & k(\mathbf{x}_*, \mathbf{x}_N)\end{bmatrix}^T, \tag{9}$$

$$\mathbf{K} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \cdots & k(\mathbf{x}_1, \mathbf{x}_N)\\ \vdots & \ddots & \vdots\\ k(\mathbf{x}_N, \mathbf{x}_1) & \cdots & k(\mathbf{x}_N, \mathbf{x}_N)\end{bmatrix}. \tag{10}$$

Using some matrix algebra, we can derive

$$f(\mathbf{x}_*) \sim \mathcal{N}\left(\mathbf{k}^T\mathbf{K}^{-1}\mathbf{f}, k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}^T\mathbf{K}^{-1}\mathbf{k}\right). \tag{11}$$

This equation provides a predictive distribution about the unobserved function value, which follows a simple Gaussian distribution.

We represent $\mu(\cdot)$ and $\sigma^2(\cdot)$ are the predicted mean and the variance, respectively, *i.e.,*

$$\mu(\mathbf{x}_*) = \mathbf{k}^T \mathbf{K}^{-1} \mathbf{f}, \tag{12}$$

$$\sigma^2(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}^T \mathbf{K}^{-1} \mathbf{k}. \tag{13}$$

Note that we can use this predictive distribution as a means of scattered data interpolation, although our goal is not interpolation. This usage is referred to as *Gaussian process regression* (GPR). See the tutorial by Ebden [2015] for this direction.

### 1.3 Covariance Hyperparameters

To predict $\mu(\cdot)$ and $\sigma^2(\cdot)$, the model hyperparameters $\boldsymbol{\theta}$ have to be determined. Here, we consider to determine them using maximum *a posteriori* (MAP) estimation, while other options (*e.g.,* maximum likelihood estimation) are also possible. Given the data $\mathcal{D}$, the model hyperparameters are determined by maximizing the posteriori distribution of $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}^{\mathrm{MAP}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \mid \mathcal{D}). \tag{14}$$

By applying Bayes' theorem, we have

$$\begin{aligned} \boldsymbol{\theta}^{\mathrm{MAP}} &= \arg\max_{\boldsymbol{\theta}} \left\{ \frac{p(\mathcal{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{D})} \right\} \\ &= \arg\max_{\boldsymbol{\theta}} p(\mathcal{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}). \end{aligned} \tag{15}$$

From the definition of the GP prior, the conditional probability $p(\mathcal{D} \mid \boldsymbol{\theta})$ follows

$$p(\mathcal{D} \mid \boldsymbol{\theta}) = \mathcal{N}(\mathbf{f}; \mathbf{0}, \mathbf{K}). \tag{16}$$

The probability $p(\boldsymbol{\theta})$ is an arbitrary prior distribution of $\boldsymbol{\theta}$. In this study, we assume log-normal distributions for each hyperparameter:

$$p(\theta_i) = \begin{cases} \mathcal{LN}(\ln 0.500, 0.10) & (i = 1, \ldots, d+1) \\ \mathcal{LN}(\ln 0.005, 0.10) & (i = d+2) \end{cases}. \tag{17}$$

Thus, we have

$$p(\boldsymbol{\theta}) = \prod_{i=1}^{d+2} p(\theta_i). \tag{18}$$

As the gradient of the objective function in Equation 15 can be expressed in closed form (see [Rasmussen and Williams 2006]), this maximization can be efficiently performed by using standard gradient-based optimization techniques, *e.g.,* L-BFGS [Liu and Nocedal 1989].

### 1.4 Acquisition Function

So far, we have discussed computational tools for predicting unobserved function values. By using them, the next sampling point is chosen. Intuitively, we want to choose the next sampling point so that it is likely to have a larger value (since we want to find the maximum) and at the same time its evaluation is more informative (*e.g.,* visiting a point that is very close to already visited points should be less useful). To realize such properties, researchers have proposed several types of acquisition function for choosing the next sampling point, including

- *probability of improvement* (PI),
- *expected improvement* (EI), and
- *Gaussian process upper confidence bound* (GP-UCB).

See [Shahriari et al. 2016] for detailed discussions. Among them, we adopt the EI criterion [Jones et al. 1998; Mockus 1974], following the previous works [Brochu et al. 2010a, 2007].

Let $f^+$ be the maximum value among the currently observed data. The acquisition function based on EI is defined as

$$a^{\mathrm{EI}}(\mathbf{x}; \mathcal{D}) = \mathbb{E}_f[\max\{f(\mathbf{x}) - f^+, 0\}], \tag{19}$$

where $f(\cdot)$ is considered as a probabilistic variable that depends on the data $\mathcal{D}$. After some integral calculations, this can be analytically expressed in closed form as

$$a^{\mathrm{EI}}(\mathbf{x}; \mathcal{D}) = (f^+ - \mu(\mathbf{x}))\Phi(\gamma(\mathbf{x})) + \sigma(\mathbf{x})\mathcal{N}(\gamma(\mathbf{x}); 0, 1), \tag{20}$$

where $\gamma(\mathbf{x}) = (f^+ - \mu(\mathbf{x}))/\sigma(\mathbf{x})$, $\mu(\cdot)$ and $\sigma(\cdot)$ are the ones calculated in Equation 12 and Equation 13 using the MAP-estimated model hyperparameters $\boldsymbol{\theta}^{\mathrm{MAP}}$, and $\Phi(\cdot)$ is the cumulative distribution function of the standard normal. Since $a^{\mathrm{EI}}(\cdot)$ can have multiple local maximums, we use the DIRECT algorithm [Jones et al. 1993], which is a global optimization algorithm, to solve the maximization of this acquisition function.

### 1.5 Example Optimization Sequences

Figure 1 shows example sequences of applying Bayesian optimization to one-dimensional test functions. Intuitively, the next sampling point $\mathbf{x}^{\mathrm{next}}$ is selected such that both $\mu(\mathbf{x}^{\mathrm{next}})$ and $\sigma(\mathbf{x}^{\mathrm{next}})$ are large. Note that we do not intend that $\mu(\cdot)$ eventually converges to $f(\cdot)$ because this is not a regression but an optimization. For example, some regions remain uncertain (*i.e.,* having large $\sigma(\cdot)$ values) but are not sampled even after several iterations; this is because they are unlikely to contain the maximum. On the other hand, $\mathbf{x}^+$ is expected to converge to the maximum.

## 2 BAYESIAN OPTIMIZATION BASED ON LINE SEARCH ORACLE

### 2.1 Example Optimization Sequence

Figure 2 shows an example sequence (that is longer than the figure in the main paper) of optimizing a 2-dimensional test function using our Bayesian optimization based on line search oracle. It shows that we could obtain a good solution after 4 or 5 iterations in this example.

### 2.2 Discussions on Hyperparameters

We derive the model hyperparameters $\boldsymbol{\theta}$ using maximum *a posterior* (MAP) estimation. Brochu *et al.* [2007] used *expert set* hyperparameters; they manually tuned the hyperparameters and fixed them during optimization sequences. Later, Brochu *et al.* [2010a] introduced a new strategy for setting the hyperparamters by using *particle filter*; however, it requires prior data. Snoek *et al.* [2012] introduced a *fully-Bayesian* approach for computing the acquisition function; their method integrates the acquisition function over all the possible hyperparameters. We implemented the MAP, expert set, and fully-Bayesian approaches, and compared in several synthetic settings. We found that the MAP approach works better on the whole, so that we chose it.

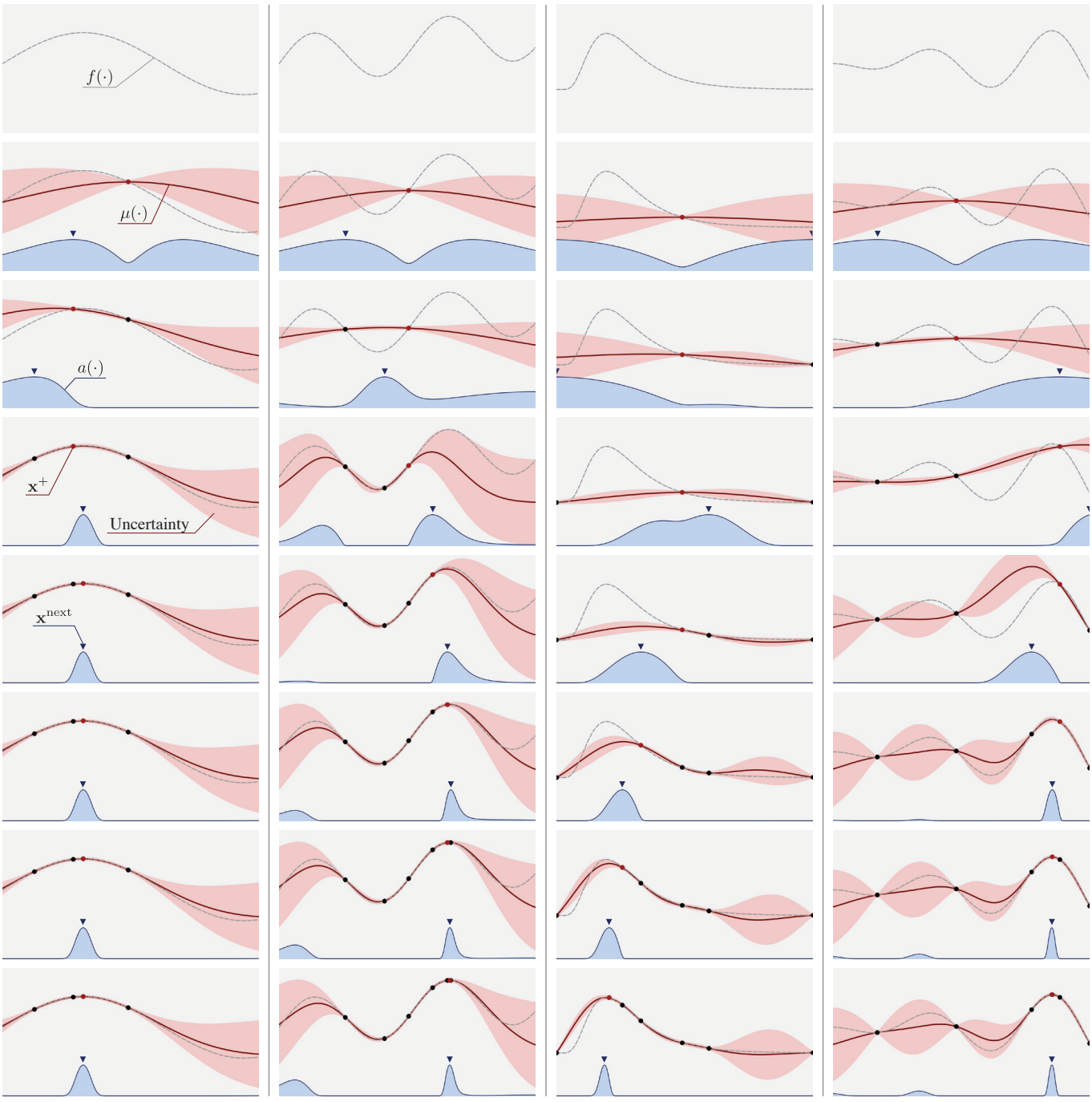


Fig. 1. **Example sequences of Bayesian optimization, applied to one-dimensional test functions.** Optimization proceeds from top to bottom. The gray dotted line indicates the unknown black-box function $f(\cdot)$, the red line indicates the predicted mean function $\mu(\cdot)$, the blue line indicates the acquisition function $a(\cdot)$, the pink region indicates the 95% confidence interval (*i.e.,* $[\mu(\cdot) - 1.96\sigma(\cdot), \mu(\cdot) + 1.96\sigma(\cdot)]$), and the dots indicate the observed data (the red one is the maximum at each moment). Note that $a(\cdot)$ is scaled for visualization purpose.

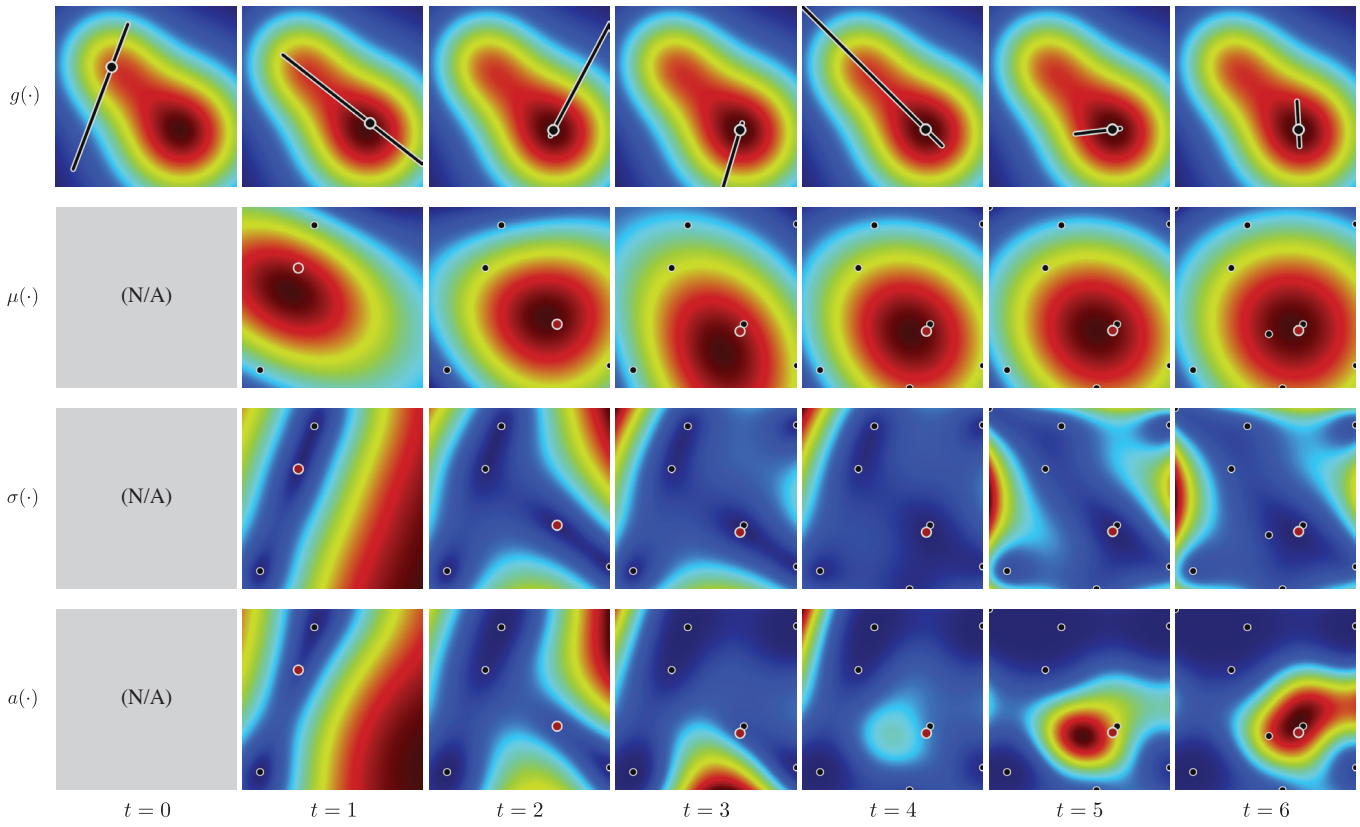


Fig. 2. **An example sequence of the Bayesian optimization based on line search oracle, applied to a two-dimensional test function.** The iteration proceeds from left to right. From top to bottom, each row visualizes the black-box function $g(\cdot)$ along with the slider space $\mathcal{S}$ and the chosen parameter set $\mathbf{x}^{\text{chosen}}$, the predicted mean function $\mu(\cdot)$, the predicted standard deviation $\sigma(\cdot)$, and the acquisition function $a(\cdot)$, respectively. The red dots denote the best parameter sets $\mathbf{x}^+$ among the observed data points at each step.

## REFERENCES


Eric Brochu, Tyson Brochu, and Nando de Freitas. 2010a. A Bayesian Interactive Optimization Approach to Procedural Animation Design. In *Proc. SCA '10*. 103–112. DOI: https://doi.org/10.2312/SCA/SCA10/103-112

Eric Brochu, Vlad M. Cora, and Nando de Freitas. 2010b. A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning. (2010). arXiv:1012.2599.

Eric Brochu, Nando de Freitas, and Abhijeet Ghosh. 2007. Active Preference Learning with Discrete Choice Data. In *Proc. NIPS '07*. 409–416. http://papers.nips.cc/paper/3219-active-preference-learning-with-discrete-choice-data.pdf

Mark Ebden. 2015. Gaussian Processes: A Quick Introduction. (2015). arXiv:1505.02965.

Donald R. Jones, Cary D. Perttunen, and Bruce E. Stuckman. 1993. Lipschitzian Optimization Without the Lipschitz Constant. *J. Optim. Theory Appl.* 79, 1 (Oct. 1993), 157–181. DOI: https://doi.org/10.1007/BF00941892

Donald R. Jones, Matthias Schonlau, and William J. Welch. 1998. Efficient Global Optimization of Expensive Black-Box Functions. *J. of Global Optimization* 13, 4 (Dec. 1998), 455–492. DOI: https://doi.org/10.1023/A:1008306431147

Dong C. Liu and Jorge Nocedal. 1989. On the Limited Memory BFGS Method for Large Scale Optimization. *Math. Program.* 45, 3 (Dec. 1989), 503–528. DOI: https://doi.org/10.1007/BF01589116

Jonas Mockus. 1974. On Bayesian Methods for Seeking the Extremum. In *Proc. IFIP Technical Conference on Optimization Techniques '74*. 400–404. DOI: https://doi.org/10.1007/3-540-07165-2_55

Carl Edward Rasmussen and Christopher K. I. Williams. 2006. *Gaussian Processes for Machine Learning*. The MIT Press.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. 2016. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proc. IEEE* 104, 1 (Jan. 2016), 148–175. DOI: https://doi.org/10.1109/JPROC.2015.2494218

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. 2012. Practical Bayesian Optimization of Machine Learning Algorithms. In *Proc. NIPS '12*. 2951–2959. http://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms.pdf