Adversarially Robust Optimization with Gaussian Processes

Ilija Bogunovic
LIONS, EPFL
ilija.bogunovic@epfl.ch

Jonathan Scarlett
National University of Singapore
scarlett@comp.nus.edu.sg

Stefanie Jegelka
MIT CSAIL
stefje@mit.edu

Volkan Cevher
LIONS, EPFL
volkan.cevher@epfl.ch

Abstract

In this paper, we consider the problem of Gaussian process (GP) optimization with an added robustness requirement: The returned point may be perturbed by an adversary, and we require the function value to remain as high as possible even after this perturbation. This problem is motivated by settings in which the underlying functions during optimization and implementation stages are different, or when one is interested in finding an entire region of good inputs rather than only a single point. We show that standard GP optimization algorithms do not exhibit the desired robustness properties, and provide a novel confidence-bound based algorithm StableOpt for this purpose. We rigorously establish the required number of samples for StableOpt to find a near-optimal point, and we complement this guarantee with an algorithm-independent lower bound. We experimentally demonstrate several potential applications of interest using real-world data sets, and we show that StableOpt consistently succeeds in finding a stable maximizer where several baseline methods fail.

1 Introduction

Gaussian processes (GP) provide a powerful means for sequentially optimizing a black-box function f that is costly to evaluate and for which noisy point evaluations are available. Since its introduction, this approach has successfully been applied to numerous applications, including robotics [21], hyperparameter tuning [30], recommender systems [34], environmental monitoring [31], and more.

In many such applications, one is faced with various forms of uncertainty that are not accounted for by standard algorithms. In robotics, the optimization is often performed via simulations, creating a mismatch between the assumed function and the true one; in hyperparameter tuning, the function is typically similarly mismatched due to limited training data; in recommendation systems and several other applications, the underlying function is inherently time-varying, so the returned solution may become increasingly stale over time; the list goes on.

In this paper, we address these considerations by studying the GP optimization problem with an additional requirement of *adversarial robustness*: The returned point may be perturbed by an adversary, and we require the function value to remain as high as possible even after this perturbation. This problem is of interest not only for attaining improved robustness to uncertainty, but also for settings where one seeks a region of good points rather than a single point, and for other related max-min optimization settings (see Section 4 for further discussion).

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

Related work. Numerous algorithms have been developed for GP optimization in recent years [7, 16, 17, 26, 28, 31, 35]. Beyond the standard setting, several important extensions have been considered, including batch sampling [11, 12, 14], contextual and time-varying settings [6, 20], safety requirements [33], and high dimensional settings [18, 25, 36], just to name a few.

Various forms of robustness in GP optimization have been considered previously. A prominent example is that of outliers [22], in which certain function values are highly unreliable; however, this is a separate issue from that of the present paper, since in [22] the returned point does not undergo any perturbation. Another related recent work is [2], which assumes that the *sampled points* (rather than the returned one) are subject to uncertainty. In addition to this difference, the uncertainty in [2] is random rather than adversarial, which is complementary but distinct from our work. The same is true of a setting called *unscented Bayesian optimization* in [23]. Moreover, no theoretical results are given in [2, 23]. In [8], a robust form of batch optimization is considered, but with yet another form of robustness, namely, some experiments in the batch may fail to produce an outcome. Level-set estimation [7, 15] is another approach to finding regions of good points rather than a single point.

Our problem formulation is also related to other works on non-convex robust optimization, particularly those of *Bertsimas et al.* [3, 4]. In these works, a stable design $\mathbf{x}$ is sought that solves $\min_{\mathbf{x}\in D}\max_{\boldsymbol{\delta}\in\mathcal{U}} f(\mathbf{x}+\boldsymbol{\delta})$. Here, $\boldsymbol{\delta}$ resides in some uncertainty set $\mathcal{U}$, and represents the perturbation against which the design $\mathbf{x}$ needs to be protected. Related problems have also recently been considered in the context of adversarial training (e.g., [29]). Compared to these works, our work bears the crucial difference that the objective function is *unknown*, and we can only learn about it through noisy point evaluations (i.e. bandit feedback).

Other works, such as [5, 9, 19, 32, 37], have considered robust optimization problems of the following form: For a given set of objectives $\{f_1, \dots, f_m\}$ find $\mathbf{x}$ achieving $\max_{\mathbf{x}\in D}\min_{i=1,\dots,m} f_i(\mathbf{x})$. We discuss variations of our algorithm for this type of formulation in Section 4.

Contributions. We introduce a variant of GP optimization in which the returned solution is required to exhibit stability/robustness to an adversarial perturbation. We demonstrate the failures of standard algorithms, and introduce a new algorithm StableOpt that overcomes these limitations. We provide a novel theoretical analysis characterizing the number of samples required for StableOpt to attain a near-optimal robust solution, and we complement this with an algorithm-independent lower bound. We provide several variations of our max-min optimization framework and theory, including connections and comparisons to previous works. Finally, we experimentally demonstrate a variety of potential applications of interest using real-world data sets, and we show that StableOpt consistently succeeds in finding a stable maximizer where several baseline methods fail.

2 Problem Setup

Model. Let f be an unknown reward function over a domain $D \subseteq \mathbb{R}^p$ for some dimension p. At time t, we query f at a single point $\mathbf{x}_t \in D$ and observe a noisy sample $y_t = f(\mathbf{x}_t) + z_t$, where $z_t \sim \mathcal{N}(0, \sigma^2)$. After T rounds, a recommended point $\mathbf{x}^{(T)}$ is returned. In contrast with the standard goal of making $f(\mathbf{x}^{(T)})$ as high as possible, we seek to find a point such that f remains high even after an adversarial perturbation; a formal description is given below.

We assume that D is endowed with a kernel function $k(\cdot,\cdot)$, and f has a bounded norm in the corresponding Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_k(D)$. Specifically, we assume that $f \in \mathcal{F}_k(B)$, where

$$\mathcal{F}_k(B) = \{f \in \mathcal{H}_k(D) \,:\, \|f\|_k \le B\}, \tag{1}$$

and $\|f\|_k$ is the RKHS norm in $\mathcal{H}_k(D)$. It is well-known that this assumption permits the construction of confidence bounds via Gaussian process (GP) methods; see Lemma 1 below for a precise statement. We assume that the kernel is normalized to satisfy $k(\mathbf{x},\mathbf{x}) = 1$ for all $\mathbf{x} \in D$. Two commonly-considered kernels are squared exponential (SE) and Matérn:

$$k_{\text{SE}}(\mathbf{x},\mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2l^2}\right), \tag{2}$$

$$k_{\text{Mat}}(\mathbf{x},\mathbf{x}') = \frac{2^{1-\nu}}{\Gamma(\nu)}\Big(\frac{\sqrt{2\nu}\|\mathbf{x}-\mathbf{x}'\|}{l}\Big) J_\nu\Big(\frac{\sqrt{2\nu}\|\mathbf{x}-\mathbf{x}'\|}{l}\Big), \tag{3}$$

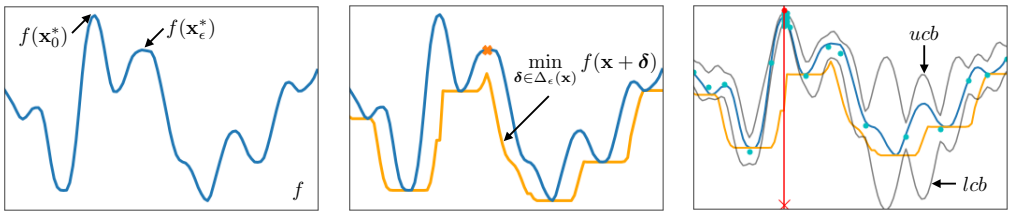


Figure 1: (Left) A function f and its maximizer $\mathbf{x}_0^*$. (Middle) For $\epsilon = 0.06$ and $d(x, x') = |x - x'|$, the decision that corresponds to the local "wider" maximum of f is the *optimal ϵ-stable* decision. (Right) GP-UCB selects a point that nearly maximizes f, but is suboptimal in the ϵ-stable sense.

where l denotes the length-scale, $\nu > 0$ is an additional parameter that dictates the smoothness, and $J(\nu)$ and $\Gamma(\nu)$ denote the modified Bessel function and the gamma function, respectively [24].

Given a sequence of decisions $\{\mathbf{x}_1, \cdots, \mathbf{x}_t\}$ and their noisy observations $\{y_1, \cdots, y_t\}$, the posterior distribution under a $\mathrm{GP}(0, k(\mathbf{x}, \mathbf{x}'))$ prior is also Gaussian, with the following mean and variance:

$$\mu_t(\mathbf{x}) = \mathbf{k}_t(\mathbf{x})^T \left(\mathbf{K}_t + \sigma^2 \mathbf{I}\right)^{-1} \mathbf{y}_t, \tag{4}$$

$$\sigma_t^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}_t(\mathbf{x})^T \left(\mathbf{K}_t + \sigma^2 \mathbf{I}\right)^{-1} \mathbf{k}_t(\mathbf{x}), \tag{5}$$

where $\mathbf{k}_t(\mathbf{x}) = \left[k(\mathbf{x}_i, \mathbf{x})\right]_{i=1}^t$, and $\mathbf{K}_t = \left[k(\mathbf{x}_t, \mathbf{x}_{t'})\right]_{t,t'}$ is the kernel matrix.

Optimization goal. Let $d(\mathbf{x}, \mathbf{x}')$ be a function mapping $D \times D \to \mathbb{R}$, and let ϵ be a constant known as the *stability parameter*. For each point $\mathbf{x} \in D$, we define a set

$$\Delta_\epsilon(\mathbf{x}) = \left\{\mathbf{x}' - \mathbf{x} \,:\, \mathbf{x}' \in D \;\text{ and }\; d(\mathbf{x}, \mathbf{x}') \le \epsilon\right\}. \tag{6}$$

One can interpret this as the set of perturbations of $\mathbf{x}$ such that the newly obtained point $\mathbf{x}'$ is within a "distance" ϵ of $\mathbf{x}$. While we refer to $d(\cdot, \cdot)$ as the distance function throughout the paper, we allow it to be a general function, and not necessarily a distance in the mathematical sense. As we exemplify in Section 5, the parameter ϵ might be naturally specified as part of the application, or might be better treated as a parameter that can be tuned for the purpose of the overall learning goal.

We define an *ϵ-stable optimal input* to be any $\mathbf{x}_\epsilon^*$ satisfying

$$\mathbf{x}_\epsilon^* \in \arg\max_{\mathbf{x}\in D} \min_{\boldsymbol{\delta}\in\Delta_\epsilon(\mathbf{x})} f(\mathbf{x} + \boldsymbol{\delta}). \tag{7}$$

Our goal is to report a point $\mathbf{x}^{(T)}$ that is stable in the sense of having low *ϵ-regret*, defined as

$$r_\epsilon(\mathbf{x}) = \min_{\boldsymbol{\delta}\in\Delta_\epsilon(\mathbf{x}_\epsilon^*)} f(\mathbf{x}_\epsilon^* + \boldsymbol{\delta}) - \min_{\boldsymbol{\delta}\in\Delta_\epsilon(\mathbf{x})} f(\mathbf{x} + \boldsymbol{\delta}). \tag{8}$$

Note that once $r_\epsilon(\mathbf{x}) \le \eta$ for some accuracy value $\eta \ge 0$, it follows that

$$\min_{\boldsymbol{\delta}\in\Delta_\epsilon(\mathbf{x})} f(\mathbf{x} + \boldsymbol{\delta}) \ge \min_{\boldsymbol{\delta}\in\Delta_\epsilon(\mathbf{x}_\epsilon^*)} f(\mathbf{x}_\epsilon^* + \boldsymbol{\delta}) - \eta. \tag{9}$$

We assume that $d(\cdot, \cdot)$ and ϵ are known, i.e., they are specified as part of the optimization formulation.

As a running example, we consider the case that $d(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|$ for some norm $\|\cdot\|$ (e.g., ℓ_2-norm), in which case achieving low ϵ-regret amounts to favoring *broad peaks* instead of narrow ones, particularly for higher ϵ; see Figure 1 for an illustration. In Section 4, we discuss how our framework also captures a variety of other max-min optimization settings of interest.

Failure of classical methods. Various algorithms have been developed for achieving small regret in the standard GP optimization problem. A prominent example is GP-UCB, which chooses

$$\mathbf{x}_t \in \arg\max_{\mathbf{x}\in D} \mathrm{ucb}_{t-1}(\mathbf{x}), \tag{10}$$

where $\mathrm{ucb}_{t-1}(\mathbf{x}) := \mu_{t-1}(\mathbf{x}) + \beta_t^{1/2} \sigma_{t-1}(\mathbf{x})$. This algorithm is guaranteed to achieve sublinear cumulative regret with high probability [31], for a suitably chosen β_t. While this is useful when

Algorithm 1 The STABLEOPT algorithm

Input: Domain D, GP prior (μ_0, σ_0, k), parameters $\{\beta_t\}_{t\geq 1}$, stability ϵ, distance function $d(\cdot,\cdot)$

1: **for** $t = 1, 2, \ldots, T$ **do**

2: Set

$$\tilde{\mathbf{x}}_t = \arg\max_{\mathbf{x}\in D} \min_{\boldsymbol{\delta}\in\Delta_\epsilon(\mathbf{x})} \mathrm{ucb}_{t-1}(\mathbf{x}+\boldsymbol{\delta}). \tag{13}$$

3: Set $\boldsymbol{\delta}_t = \arg\min_{\boldsymbol{\delta}\in\Delta_\epsilon(\tilde{\mathbf{x}}_t)} \mathrm{lcb}_{t-1}(\tilde{\mathbf{x}}_t+\boldsymbol{\delta})$

4: Sample $\tilde{\mathbf{x}}_t + \boldsymbol{\delta}_t$, and observe $y_t = f(\tilde{\mathbf{x}}_t + \boldsymbol{\delta}_t) + z_t$

5: Update μ_t, σ_t, ucb_t and lcb_t according to (5) and (12), by including $\{(\tilde{\mathbf{x}}_t + \boldsymbol{\delta}_t, y_t)\}$

6: **end for**

$\mathbf{x}^*_\epsilon = \mathbf{x}^*_0$,[1] in general for a given fixed $\epsilon \neq 0$, these two decisions may not coincide, and hence, $\min_{\boldsymbol{\delta}\in\Delta_\epsilon(\mathbf{x}^*_0)} f(\mathbf{x}^*_0+\boldsymbol{\delta})$ can be significantly smaller than $\min_{\boldsymbol{\delta}\in\Delta_\epsilon(\mathbf{x}^*_\epsilon)} f(\mathbf{x}^*_\epsilon+\boldsymbol{\delta})$.

A visual example is given in Figure 1 (Right), where the selected point of GP-UCB for $t = 20$ is shown. This point nearly maximizes f, but it is strictly suboptimal in the ϵ-stable sense. The same limitation applies to other GP optimization strategies (e.g., [7, 16, 17, 26, 28, 35]) whose goal is to identify the global non-robust maximum $\mathbf{x}^*_0$. In Section 5, we will see that more advanced baseline strategies also perform poorly when applied to our problem.

3 Proposed Algorithm and Theory

Our proposed algorithm, STABLEOPT, is described in Algorithm 1, and makes use of the following confidence bounds depending on an *exploration parameter* β_t (*cf.*, Lemma 1 below):

$$\mathrm{ucb}_{t-1}(\mathbf{x}) := \mu_{t-1}(\mathbf{x}) + \beta_t^{1/2}\sigma_{t-1}(\mathbf{x}), \tag{11}$$

$$\mathrm{lcb}_{t-1}(\mathbf{x}) := \mu_{t-1}(\mathbf{x}) - \beta_t^{1/2}\sigma_{t-1}(\mathbf{x}). \tag{12}$$

The point $\tilde{\mathbf{x}}_t$ defined in (13) is the one having the highest "stable" upper confidence bound. However, the queried point is not $\tilde{\mathbf{x}}_t$, but instead $\tilde{\mathbf{x}}_t + \boldsymbol{\delta}_t$, where $\boldsymbol{\delta}_t \in \Delta_\epsilon(\tilde{\mathbf{x}}_t)$ is chosen to minimize the *lower* confidence bound. As a result, the algorithm is based on two distinct principles: (i) optimism in the face of uncertainty when it comes to selecting $\tilde{\mathbf{x}}_t$; (ii) pessimism in the face of uncertainty when it comes to anticipating the perturbation of $\tilde{\mathbf{x}}_t$. The first of these is inherent to existing algorithms such as GP-UCB [31], whereas the second is unique to the adversarially robust GP optimization problem. An example illustration of STABLEOPT's execution is given in the supplementary material.

We have left the final reported point $\mathbf{x}^{(T)}$ unspecified in Algorithm 1, as there are numerous reasonable choices. The simplest choice is to simply return $\mathbf{x}^{(T)} = \tilde{\mathbf{x}}_T$, but in our theory and experiments, we will focus on $\mathbf{x}^{(T)}$ equaling the point in $\{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_T\}$ with the highest lower confidence bound.

Finding an exact solution to the optimization of the acquisition function in (13) can be challenging in practice. When D is continuous, a natural approach is to find an approximate solution using an efficient local search algorithm for robust optimization with a fully known objective function, such as that of [4].

3.1 Upper bound on ϵ-regret

Our analysis makes use of the *maximum information gain* under t noisy measurements:

$$\gamma_t = \max_{\mathbf{x}_1,\cdots,\mathbf{x}_t} \frac{1}{2}\log\det(\mathbf{I}_t + \sigma^{-2}\mathbf{K}_t), \tag{14}$$

which has been used in numerous theoretical works on GP optimization following [31].

STABLEOPT depends on the exploration parameter β_t, which determines the width of the confidence bounds. In our main result, we set β_t as in [10] and make use of the following.

[1] In this discussion, we take $d(\mathbf{x},\mathbf{x}') = \|\mathbf{x}-\mathbf{x}'\|_2$, so that $\epsilon = 0$ recovers the standard non-stable regret [31].

Lemma 1. [10] *Fix $f \in \mathcal{F}_k(B)$, and consider the sampling model $y_t = f(\mathbf{x}_t) + z_t$ with $z_t \sim \mathcal{N}(0, \sigma^2)$, with independence between times. Under the choice $\beta_t = \big(B + \sigma\sqrt{2(\gamma_{t-1} + \log\frac{e}{\xi})}\big)^2$, the following holds with probability at least $1 - \xi$:*

$$\mathrm{lcb}_{t-1}(\mathbf{x}) \le f(\mathbf{x}) \le \mathrm{ucb}_{t-1}(\mathbf{x}), \quad \forall \mathbf{x} \in D, \forall t \ge 1. \tag{15}$$

The following theorem bounds the performance of StableOpt under a suitable choice of the recommended point $\mathbf{x}^{(T)}$. The proof is given in the supplementary material.

Theorem 1. (Upper Bound) *Fix $\epsilon > 0$, $\eta > 0$, $B > 0$, $T \in \mathbb{Z}$, $\xi \in (0, 1)$, and a distance function $d(\mathbf{x}, \mathbf{x}')$, and suppose that*

$$\frac{T}{\beta_T \gamma_T} \ge \frac{C_1}{\eta^2}, \tag{16}$$

where $C_1 = 8/\log(1 + \sigma^{-2})$. For any $f \in \mathcal{F}_k(B)$, StableOpt with β_t set as in Lemma 1 achieves $r_\epsilon(\mathbf{x}^{(T)}) \le \eta$ after T rounds with probability at least $1 - \xi$, where

$$\mathbf{x}^{(T)} = \tilde{\mathbf{x}}_{t^*}, \qquad t^* = \operatorname*{arg\,max}_{t=1,\dots,T} \min_{\boldsymbol{\delta} \in \Delta_\epsilon(\tilde{\mathbf{x}}_t)} \mathrm{lcb}_{t-1}(\tilde{\mathbf{x}}_t + \boldsymbol{\delta}). \tag{17}$$

This result holds for general kernels, and for both finite and continuous D. Our analysis bounds function values according to the confidence bounds in Lemma 1 analogously to GP-UCB [31], but also addresses the non-trivial challenge of characterizing the perturbations $\boldsymbol{\delta}_t$. While we focused on the non-Bayesian RKHS setting, the proof can easily be adapted to handle the *Bayesian optimization* (BO) setting in which $f \sim \mathrm{GP}(0, k)$; see Section 4 for further discussion.

Theorem 1 can be made more explicit by substituting bounds on γ_T; in particular, $\gamma_T = O((\log T)^{p+1})$ for the SE kernel, and $\gamma_T = O(T^{\frac{p(p+1)}{2\nu + p(p+1)}} \log T)$ for the Matérn-ν kernel [31]. The former yields $T = O^*\big(\frac{1}{\eta^2}\big(\log\frac{1}{\eta}\big)^{2p}\big)$ in Theorem 1 for constant B, σ^2, and ϵ (where $O^*(\cdot)$ hides dimension-independent log factors), which we will shortly see nearly matches an algorithm-independent lower bound.

3.2 Lower bound on ϵ-regret

Establishing lower bounds under general kernels and input domains is an open problem even in the non-robust setting. Accordingly, the following theorem focuses on a more specific setting than the upper bound: We let the input domain be $[0, 1]^p$ for some dimension p, and we focus on the SE and Matérn kernels. In addition, we only consider the case that $d(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|_2$, though extensions to other norms (e.g., ℓ_1 or ℓ_∞) follow immediately from the proof.

Theorem 2. (Lower Bound) *Let $D = [0, 1]^p$ for some dimension p, and set $d(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|_2$. Fix $\epsilon \in \big(0, \frac{1}{2}\big)$, $\eta \in \big(0, \frac{1}{2}\big)$, $B > 0$, and $T \in \mathbb{Z}$. Suppose there exists an algorithm that, for any $f \in \mathcal{F}_k(B)$, reports a point $\mathbf{x}^{(T)}$ achieving ϵ-regret $r_\epsilon(\mathbf{x}^{(T)}) \le \eta$ after T rounds with probability at least $1 - \xi$. Then, provided that $\frac{\eta}{B}$ and ξ are sufficiently small, we have the following:*

1. *For $k = k_{\text{SE}}$, it is necessary that $T = \Omega\big(\frac{\sigma^2}{\eta^2}\big(\log\frac{B}{\eta}\big)^{p/2}\big)$.*
2. *For $k = k_{\text{Matérn}}$, it is necessary that $T = \Omega\big(\frac{\sigma^2}{\eta^2}\big(\frac{B}{\eta}\big)^{p/\nu}\big)$.*

Here we assume that the stability parameter ϵ, dimension p, target probability ξ, and kernel parameters l, ν are fixed (i.e., not varying as a function of the parameters T, η, σ and B).

The proof is based on constructing a finite subset of "difficult" functions in $\mathcal{F}_k(B)$ and applying lower bounding techniques from the multi-armed bandit literature, also making use of several auxiliary results from the non-robust setting [27]. More specifically, the functions in the restricted class consist of narrow negative "valleys" that the adversary can perturb the reported point into, but that are hard to identify until a large number of samples have been taken.

For constant σ^2 and B, the condition for the SE kernel simplifies to $T = \Omega\big(\frac{1}{\eta^2}\big(\log\frac{1}{\eta}\big)^{p/2}\big)$, thus nearly matching the upper bound $T = O^*\big(\frac{1}{\eta^2}\big(\log\frac{1}{\eta}\big)^{2p}\big)$ of StableOpt established above. In the case of the Matérn kernel, more significant gaps remain between the upper and lower bounds; however, similar gaps remain even in the standard (non-robust) setting [27].

4 Variations of StableOpt

While the above problem formulation seeks robustness within an ϵ-ball corresponding to the distance function $d(\cdot, \cdot)$, our algorithm and theory apply to a variety of seemingly distinct settings. We outline a few such settings here; in the supplementary material, we give details of their derivations.

Robust Bayesian optimization. Algorithm 1 and Theorem 1 extend readily to the Bayesian setting in which $f \sim \mathrm{GP}(0, k(\mathbf{x}, \mathbf{x}'))$. In particular, since the proof of Theorem 1 is based on confidence bounds, the only change required is selecting β_t to be that used for the Bayesian setting in [31]. As a result, our framework also captures the novel problem of *adversarially robust Bayesian optimization*. All of the variations outlined below similarly apply to both the Bayesian and non-Bayesian settings.

Robustness to unknown parameters. Consider the scenario where an unknown function $f : D \times \Theta \to \mathbb{R}$ has a bounded RKHS norm under some composite kernel $k((\mathbf{x}, \boldsymbol{\theta}), (\mathbf{x}', \boldsymbol{\theta}'))$. Some special cases include $k((\mathbf{x}, \boldsymbol{\theta}), (\mathbf{x}', \boldsymbol{\theta}')) = k(\mathbf{x}, \mathbf{x}') + k(\boldsymbol{\theta}, \boldsymbol{\theta}')$ and $k((\mathbf{x}, \boldsymbol{\theta}), (\mathbf{x}', \boldsymbol{\theta}')) = k(\mathbf{x}, \mathbf{x}')k(\boldsymbol{\theta}, \boldsymbol{\theta}')$ [20]. The posterior mean $\mu_t(\mathbf{x}, \boldsymbol{\theta})$ and variance $\sigma_t^2(\mathbf{x}, \boldsymbol{\theta})$ conditioned on the previous observations $(\mathbf{x}_1, \boldsymbol{\theta}_1, y_1), ..., (\mathbf{x}_{t-1}, \boldsymbol{\theta}_{t-1}, y_{t-1})$ are computed analogously to (5) [20].

A robust optimization formulation in this setting is to seek $\mathbf{x}$ that solves

$$\max_{\mathbf{x} \in D} \min_{\boldsymbol{\theta} \in \Theta} f(\mathbf{x}, \boldsymbol{\theta}). \tag{18}$$

That is, we seek to find a configuration $\mathbf{x}$ that is robust against any possible parameter vector $\boldsymbol{\theta} \in \Theta$.

Potential applications of this setup include the following:

- In areas such a robotics, we may have numerous parameters to tune (given by $\mathbf{x}$ and $\boldsymbol{\theta}$ collectively), but when it comes to implementation, some of them (i.e., only $\boldsymbol{\theta}$) become out of our control. Hence, we need to be robust against whatever values they may take.
- We wish to tune hyperparameters in order to make an algorithm work simultaneously for a number of distinct data types that bear some similarities/correlations. The data types are represented by $\boldsymbol{\theta}$, and we can choose the data type to our liking during the optimization stage.

StableOpt can be used to solve (18); we maintain $\boldsymbol{\theta}_t$ instead of $\boldsymbol{\delta}_t$, and modify the main steps to

$$\mathbf{x}_t \in \arg\max_{\mathbf{x} \in D} \min_{\boldsymbol{\theta} \in \Theta} \mathrm{ucb}_{t-1}(\mathbf{x}, \boldsymbol{\theta}), \tag{19}$$

$$\boldsymbol{\theta}_t \in \arg\min_{\boldsymbol{\theta} \in \Theta} \mathrm{lcb}_{t-1}(\mathbf{x}_t, \boldsymbol{\theta}). \tag{20}$$

At each time step, StableOpt receives a noisy observation $y_t = f(\mathbf{x}_t, \boldsymbol{\theta}_t) + z_t$, which is used with $(\mathbf{x}_t, \boldsymbol{\theta}_t)$ for computing the posterior. As explained in the supplementary material, the guarantee $r_\epsilon(\mathbf{x}^{(T)}) \leq \eta$ in Theorem 1 is replaced by $\min_{\boldsymbol{\theta} \in \Theta} f(\mathbf{x}^{(T)}, \boldsymbol{\theta}) \geq \max_{\mathbf{x} \in D} \min_{\boldsymbol{\theta} \in \Theta} f(\mathbf{x}, \boldsymbol{\theta}) - \eta$.

Robust estimation. Continuing with the consideration of a composite kernel on $(\mathbf{x}, \boldsymbol{\theta})$, we consider the following practical problem variant proposed in [4]. Let $\bar{\boldsymbol{\theta}} \in \Theta$ be an estimate of the true problem coefficient $\boldsymbol{\theta}^* \in \Theta$. Since, $\bar{\boldsymbol{\theta}}$ is an estimate, the true coefficient satisfies $\boldsymbol{\theta}^* = \bar{\boldsymbol{\theta}} + \boldsymbol{\delta}_{\boldsymbol{\theta}}$, where $\boldsymbol{\delta}_{\boldsymbol{\theta}}$ represents uncertainty in $\bar{\boldsymbol{\theta}}$. Often, practitioners solve $\max_{\mathbf{x} \in D} f(\mathbf{x}, \bar{\boldsymbol{\theta}})$ and ignore the uncertainty. As a more sophisticated approach, we let $\Delta_\epsilon(\bar{\boldsymbol{\theta}}) = \left\{\boldsymbol{\theta} - \bar{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta \text{ and } d(\bar{\boldsymbol{\theta}}, \boldsymbol{\theta}) \leq \epsilon\right\}$, and consider the following robust problem formulation:

$$\max_{\mathbf{x} \in D} \min_{\boldsymbol{\delta}_{\boldsymbol{\theta}} \in \Delta_\epsilon(\bar{\boldsymbol{\theta}})} f(\mathbf{x}, \bar{\boldsymbol{\theta}} + \boldsymbol{\delta}_{\boldsymbol{\theta}}). \tag{21}$$

For the given estimate $\bar{\boldsymbol{\theta}}$, the main steps of StableOpt in this setting are

$$\mathbf{x}_t \in \arg\max_{\mathbf{x} \in D} \min_{\boldsymbol{\delta}_{\boldsymbol{\theta}} \in \Delta_\epsilon(\bar{\boldsymbol{\theta}})} \mathrm{ucb}_{t-1}(\mathbf{x}, \bar{\boldsymbol{\theta}} + \boldsymbol{\delta}_{\boldsymbol{\theta}}), \tag{22}$$

$$\boldsymbol{\delta}_{\boldsymbol{\theta},t} \in \arg\min_{\boldsymbol{\delta}_{\boldsymbol{\theta}} \in \Delta_\epsilon(\bar{\boldsymbol{\theta}})} \mathrm{lcb}_{t-1}(\mathbf{x}_t, \bar{\boldsymbol{\theta}} + \boldsymbol{\delta}_{\boldsymbol{\theta}}), \tag{23}$$

and the noisy observations take the form $y_t = f(\mathbf{x}_t, \bar{\boldsymbol{\theta}} + \boldsymbol{\delta}_{\boldsymbol{\theta},t}) + z_t$. The guarantee $r_\epsilon(\mathbf{x}^{(T)}) \leq \eta$ in Theorem 1 is replaced by $\min_{\boldsymbol{\delta}_{\boldsymbol{\theta}} \in \Delta_\epsilon(\bar{\boldsymbol{\theta}})} f(\mathbf{x}^{(T)}, \bar{\boldsymbol{\theta}} + \boldsymbol{\delta}_{\boldsymbol{\theta}}) \geq \max_{\mathbf{x} \in D} \min_{\boldsymbol{\delta}_{\boldsymbol{\theta}} \in \Delta_\epsilon(\bar{\boldsymbol{\theta}})} f(\mathbf{x}, \bar{\boldsymbol{\theta}} + \boldsymbol{\delta}_{\boldsymbol{\theta}}) - \eta$.

Robust group identification. In some applications, it is natural to partition D into disjoint subsets, and search for the subset with the highest worst-case function value (see Section 5 for a movie

recommendation example). Letting $\mathcal{G} = \{G_1, \ldots, G_k\}$ denote the groups that partition the input space, the robust optimization problem is given by

$$\max_{G \in \mathcal{G}} \min_{\mathbf{x} \in G} f(\mathbf{x}), \tag{24}$$

and the algorithm reports a group $G^{(T)}$. The main steps of StableOpt are given by

$$G_t \in \arg\max_{G \in \mathcal{G}} \min_{\mathbf{x} \in G} \mathrm{ucb}_{t-1}(\mathbf{x}), \tag{25}$$

$$\mathbf{x}_t \in \arg\min_{\mathbf{x} \in G_t} \mathrm{lcb}_{t-1}(\mathbf{x}), \tag{26}$$

and the observations are of the form $y_t = f(\mathbf{x}_t) + z_t$ as usual. The guarantee $r_\epsilon(\mathbf{x}^{(T)}) \leq \eta$ in Theorem 1 is replaced by $\min_{\mathbf{x} \in G^{(T)}} f(\mathbf{x}) \geq \max_{G \in \mathcal{G}} \min_{\mathbf{x} \in G} f(\mathbf{x}) - \eta$.

The preceding variations of StableOpt, as well as their resulting variations of Theorem 1, follow by substituting certain (unconventional) choices of $d(\cdot, \cdot)$ and ϵ into Algorithm 1 and Theorem 1, with $(\mathbf{x}, \boldsymbol{\theta})$ in place of $\mathbf{x}$ where appropriate. The details are given in the supplementary material.

5 Experiments

In this section, we experimentally validate the performance of StableOpt by comparing against several baselines. Each algorithm that we consider may distinguish between the *sampled point* (i.e., the point that produces the noisy observation y_t) and the *reported point* (i.e., the point believed to be near-optimal in terms of ϵ-stability). For StableOpt, as described in Algorithm 1, the sampled point is $\tilde{\mathbf{x}}_t + \boldsymbol{\delta}_t$, and the reported point after time t is the one in $\{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_t\}$ with the highest value of $\min_{\boldsymbol{\delta} \in \Delta_\epsilon(\tilde{\mathbf{x}}_t)} \mathrm{lcb}_t(\tilde{\mathbf{x}}_t + \boldsymbol{\delta})$.[2] In addition, we consider the following baselines:

- GP-UCB (see (10)). We consider GP-UCB to be a good representative of the wide range of existing methods that search for the non-robust global maximum.
- MaxiMin-GP-UCB. We consider a natural generalization of GP-UCB in which, at each time step, the sampled and reported point are both given by

$$\mathbf{x}_t = \arg\max_{\mathbf{x} \in D} \min_{\boldsymbol{\delta} \in \Delta_\epsilon(x)} \mathrm{ucb}_{t-1}(\mathbf{x} + \boldsymbol{\delta}). \tag{27}$$

- Stable-GP-Random. The sampling point $\mathbf{x}_t$ at every time step is chosen uniformly at random, while the reported point at time t is chosen to be the point among the sampled points $\{\mathbf{x}_1, \ldots, \mathbf{x}_t\}$ according to the same rule as the one used for StableOpt.
- Stable-GP-UCB. The sampled point is given by the GP-UCB rule, while the reported point is again chosen in the same way as in StableOpt.

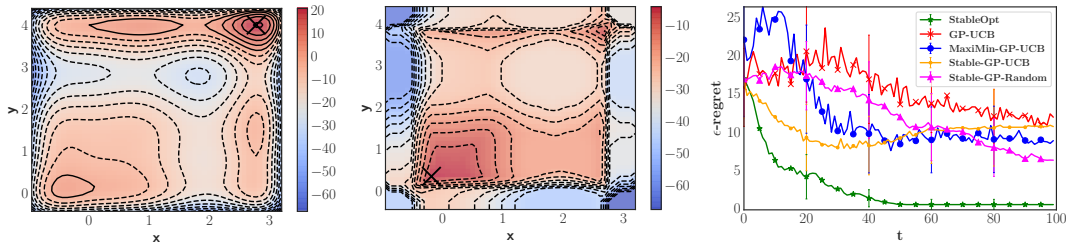
As observed in existing works (e.g., [7, 31]), the theoretical choice of β_t is overly conservative. We therefore adopt a constant value of $\beta_t^{1/2} = 2.0$ in each of the above methods, which we found to provide a suitable exploration/exploitation trade-off for each of the above algorithms.

Given a reported point $\mathbf{x}^{(t)}$ at time t, the performance metric is the ϵ-regret $r_\epsilon(\mathbf{x}^{(t)})$ given in (8). Two observations are in order: (i) All the baselines are heuristic approaches for our problem, and they do not have guarantees in terms of near-optimal stability; (ii) We do not compare against other standard GP optimization methods, as their performance is comparable to that of GP-UCB; in particular, they suffer from exactly the same pitfalls described at the end of Section 2.

Synthetic function. We consider the synthetic function from [4] (see Figure 2a), given by

$$\begin{aligned} f_{\mathrm{poly}}(x, y) = &-2x^6 + 12.2x^5 - 21.2x^4 - 6.2x + 6.4x^3 + 4.7x^2 - y^6 + 11y^5 \\ &- 43.3y^4 + 10y + 74.8y^3 - 56.9y^2 + 4.1xy + 0.1y^2x^2 - 0.4y^2x - 0.4x^2y. \end{aligned} \tag{28}$$

[2] This is slightly different from Theorem 1, which uses the confidence bound $\mathrm{lcb}_{\tau-1}$ for $\mathbf{x}_\tau$ instead of adopting the common bound lcb_t. We found the latter to be more suitable when the kernel hyperparameters are updated online, whereas Theorem 1 assumes a known kernel. Theorem 1 can be adapted to use lcb_t alone by intersecting the confidence bounds at each time instant so that they are monotonically shrinking [15].



(a) $f_{\text{poly}}(x, y)$ (b) $g_{\text{poly}}(x, y)$ (c) ϵ-regret

Figure 2: (Left) Synthetic function from [4]. (Middle) Counterpart with worst-case perturbations. (Right) The performance. In this example, StableOpt significantly outperforms the baselines.

The decision space is a uniformly spaced grid of points in $((-0.95, 3.2), (-0.45, 4.4))$ of size 10^4. There exist multiple local maxima, and the global maximum is at $(x_f^*, y_f^*) = (2.82, 4.0)$, with $f_{\text{poly}}(x_f^*, y_f^*) = 20.82$. Similarly as in [4], given stability parameters $\boldsymbol{\delta} = (\delta_x, \delta_y)$, where $\|\boldsymbol{\delta}\|_2 \leq 0.5$, the robust optimization problem is

$$\max_{(x,y)\in D} g_{\text{poly}}(x, y), \tag{29}$$

where

$$g_{\text{poly}}(x, y) := \min_{(\delta_x,\delta_y)\in\Delta_{0.5}(x,y)} f_{\text{poly}}(x - \delta_x, y - \delta_y). \tag{30}$$

A plot of g_{poly} is shown in Figure 2b. The global maximum is attained at $(x_g^*, y_g^*) = (-0.195, 0.284)$ and $g_{\text{poly}}(x_g^*, y_g^*) = -4.33$, and the inputs maximizing f yield $g_{\text{poly}}(x_f^*, y_f^*) = -22.34$.

We initialize the above algorithms by selecting 10 uniformly random inputs (x, y), keeping those points the same for each algorithm. The kernel adopted is a squared exponential ARD kernel. We randomly sample 500 points whose function value is above -15.0 to learn the GP hyperparameters via maximum likelihood, and then run the algorithms with these hyperparameters. The observation noise standard deviation is set to 0.1, and the number of sampling rounds is $T = 100$. We repeat the experiment 100 times and show the average performance in Figure 2c. We observe that StableOpt significantly outperforms the baselines in this experiment. In the later rounds, the baselines report points that are close to the global optimizer, which is suboptimal with respect to the ϵ-regret.

Lake data. In the supplementary material, we provide an analogous experiment to that above using chlorophyll concentration data from Lake Zürich, with StableOpt again performing best.

Robust robot pushing. We consider the deterministic version of the robot pushing objective from [35], with publicly available code.[3] The goal is to find a good pre-image for pushing an object to a target location. The 3-dimensional function takes as input the robot location (r_x, r_y) and pushing duration r_t, and outputs $f(r_x, r_y, r_t) = 5 - d_{\text{end}}$, where d_{end} is the distance from the pushed object to the target location. The domain D is continuous: $r_x, r_y \in [-5, 5]$ and $r_t \in [1, 30]$.

We consider a twist on this problem in which there is uncertainty regarding the precise target location, so one seeks a set of input parameters that is robust against a number of different potential locations. In the simplest case, the number of such locations is finite, meaning we can model this problem as $\boldsymbol{r} \in \arg\max_{\boldsymbol{r}\in D} \min_{i\in[m]} f_i(\boldsymbol{r})$, where each f_i corresponds to a different target location, and $[m] = \{1, \ldots, m\}$. This is a special case of (18) with a finite set Θ of cardinality m.

In our experiment, we use $m = 2$. Hence, our goal is to find an input configuration $\boldsymbol{r}$ that is robust against two different target locations. The first target is uniform over the domain, and the second is uniform over the ℓ_1-ball centered at the first target location with radius $r = 2.0$. We initialize each algorithm with one random sample from each f_i. We run each method for $T = 100$ rounds, and for a reported point $\boldsymbol{r}_t$ at time t, we compare the methods in terms of the robust objective $\min_{i\in[m]} f_i(\boldsymbol{r}_t)$. We perform a fully Bayesian treatment of the hyperparameters, sampling every 10 rounds as in [17, 35]. We average over 30 random pairs of $\{f_1, f_2\}$ and report the results in Figure 3. StableOpt noticeably outperforms its competitors except in some of the very early rounds. We note that the apparent discontinuities in certain curves are a result of the hyperparmeter re-estimation.

[3] https://github.com/zi-w/Max-value-Entropy-Search

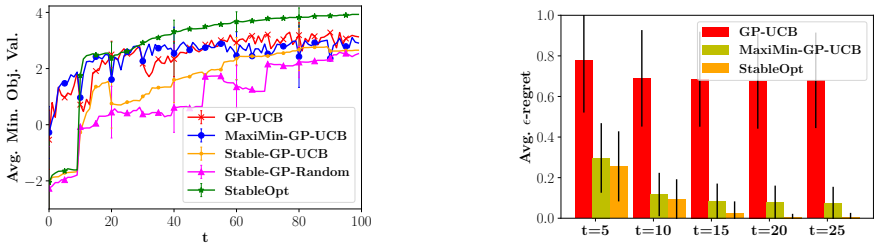


Figure 3: Robust robot pushing experiment (Left) and MovieLens-100K experiment (Right)

Group movie recommendation. Our goal in this task is to recommend a group of movies to a user such that *every* movie in the group is to their liking. We use the MovieLens-100K dataset, which consists of 1682 movies and 943 users. The data takes the form of an incomplete matrix $\mathbf{R}$ of ratings, where $R_{i,j}$ is the rating of movie i given by the user j. To impute the missing rating values, we apply non-negative matrix factorization with $p = 15$ latent factors. This produces a feature vector for each movie $\mathbf{m}_i \in \mathbb{R}^p$ and user $\mathbf{u}_j \in \mathbb{R}^p$. We use 10% of the user data for training, in which we fit a Gaussian distribution $P(\mathbf{u}) = \mathcal{N}(\mathbf{u}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$. For a given user $\mathbf{u}_j$ in the test set, $P(\mathbf{u})$ is considered to be a prior, and the objective is given by $f_j(\mathbf{m}_i) = \mathbf{m}_i^T \mathbf{u}_j$, corresponding to a GP with a linear kernel.

We cluster the movie feature vectors into $k = 80$ groups, i.e., $\mathcal{G} = \{G_1, \ldots, G_k\}$, via the k-means algorithm. Similarly to (26), the robust optimization problem for a given user j is

$$\max_{G \in \mathcal{G}} g_j(G), \tag{31}$$

where $g_j(G) = \min_{\mathbf{m}_i \in G} f_j(\mathbf{m}_i)$. That is, for the user with feature vector $\mathbf{u}_j$, our goal is to find the group of movies to recommend such that the entire collection of movies is robust with respect to the user's preferences.

In this experiment, we compare StableOpt against GP-UCB and MaxiMin-GP-UCB. We report the ϵ-regret given by $g_j(G^*) - g_j(G^{(t)})$ where G^* is the maximizer of (31), and $G^{(t)}$ is the reported group after time t. Since we are reporting groups rather than points, the baselines require slight modifications: At time t, GP-UCB selects the movie $\mathbf{m}_t$ (i.e., asks for the user's rating of it) and reports the group $G^{(t)}$ to which $\mathbf{m}_t$ belongs. MaxiMin-GP-UCB reports $G^{(t)} \in \arg\max_{G \in \mathcal{G}} \min_{\mathbf{m} \in G} \mathrm{ucb}_{t-1}(\mathbf{m})$ and then selects $\mathbf{m}_t \in \arg\min_{\mathbf{m} \in G^{(t)}} \mathrm{ucb}_{t-1}(\mathbf{m})$. Finally, StableOpt reports a group in the same way as MaxiMin-GP-UCB, but selects $\mathbf{m}_t$ analogously to (26). In Figure 3, we show the average ϵ-regret, where the average is taken over 500 different test users. In this experiment, the average ϵ-regret decreases rapidly after only a small number of rounds. Among the three methods, StableOpt is the only one that finds the optimal movie group.

6 Conclusion

We have introduced and studied a variant of GP optimization in which one requires stability/robustness to an adversarial perturbation. We demonstrated the failures of existing algorithms, and provided a new algorithm StableOpt that overcomes these limitations, with rigorous guarantees. We showed that our framework naturally applies to several interesting max-min optimization formulations, and we demonstrated significant improvements over some natural baselines in the experimental examples.

An interesting direction for future work is to study the ϵ-stable optimization formulation in the context of hyperparameter tuning (e.g., for deep neural networks). One might expect that wide function maxima in hyperparameter space provide better generalization than narrow maxima, but establishing this requires further investigation. Similar considerations are an ongoing topic of debate in understanding the *parameter space* rather than the hyperparmeter space, e.g., see [13].

Acknowledgment. This work was partially supported by the Swiss National Science Foundation (SNSF) under grant number 407540_167319, by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no725594 - time-data), by DARPA DSO's Lagrange program under grant FA86501827838, and by an NUS startup grant.

References

[1] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. Technical report, http://www.dklevine.com/archive/refs4462.pdf, 1998.

[2] Justin J. Beland and Prasanth B. Nair. Bayesian optimization under uncertainty. NIPS BayesOpt 2017 workshop, 2017.

[3] Dimitris Bertsimas, Omid Nohadani, and Kwong Meng Teo. Nonconvex robust optimization for problems with constraints. *INFORMS Journal on Computing*, 22(1):44–58, 2010.

[4] Dimitris Bertsimas, Omid Nohadani, and Kwong Meng Teo. Robust optimization for unconstrained simulation-based problems. *Operations Research*, 58(1):161–178, 2010.

[5] Ilija Bogunovic, Slobodan Mitrović, Jonathan Scarlett, and Volkan Cevher. Robust submodular maximization: A non-uniform partitioning approach. In *International Conference on Machine Learning (ICML)*, pages 508–516, 2017.

[6] Ilija Bogunovic, Jonathan Scarlett, and Volkan Cevher. Time-varying Gaussian process bandit optimization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 314–323, 2016.

[7] Ilija Bogunovic, Jonathan Scarlett, Andreas Krause, and Volkan Cevher. Truncated variance reduction: A unified approach to Bayesian optimization and level-set estimation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1507–1515, 2016.

[8] Ilija Bogunovic, Junyao Zhao, and Volkan Cevher. Robust maximization of non-submodular objectives. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 890–899, 2018.

[9] Robert S Chen, Brendan Lucier, Yaron Singer, and Vasilis Syrgkanis. Robust optimization for non-convex objectives. In *Advances in Neural Information Processing Systems (NIPS)*, pages 4708–4717, 2017.

[10] Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning (ICML)*, pages 844–853, 2017.

[11] Emile Contal, David Buffoni, Alexandre Robicquet, and Nicolas Vayatis. Parallel Gaussian process optimization with upper confidence bound and pure exploration. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 225–240. Springer, 2013.

[12] Thomas Desautels, Andreas Krause, and Joel W Burdick. Parallelizing exploration-exploitation tradeoffs in Gaussian process bandit optimization. *Journal of Machine Learning Research*, 15(1):3873–3923, 2014.

[13] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning (ICML)*, 2017.

[14] Javier González, Zhenwen Dai, Philipp Hennig, and Neil Lawrence. Batch Bayesian optimization via local penalization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 648–657, 2016.

[15] Alkis Gotovos, Nathalie Casati, Gregory Hitz, and Andreas Krause. Active learning for level set estimation. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1344–1350, 2013.

[16] Philipp Hennig and Christian J Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(Jun):1809–1837, 2012.

[17] José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems (NIPS)*, pages 918–926, 2014.

[18] Kirthevasan Kandasamy, Jeff Schneider, and Barnabás Póczos. High dimensional Bayesian optimisation and bandits via additive models. In *International Conference on Machine Learning (ICML)*, pages 295–304, 2015.

[19] Andreas Krause, H Brendan McMahan, Carlos Guestrin, and Anupam Gupta. Robust submodular observation selection. *Journal of Machine Learning Research*, 9(Dec):2761–2801, 2008.

[20] Andreas Krause and Cheng S Ong. Contextual Gaussian process bandit optimization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2447–2455, 2011.

[21] Daniel J Lizotte, Tao Wang, Michael H Bowling, and Dale Schuurmans. Automatic gait optimization with Gaussian process regression. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 944–949, 2007.

[22] Ruben Martinez-Cantin, Kevin Tee, and Michael McCourt. Practical Bayesian optimization in the presence of outliers. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018.

[23] J. Nogueira, R. Martinez-Cantin, A. Bernardino, and L. Jamone. Unscented Bayesian optimization for safe robot grasping. In *IEEE/RSJ Int. Conf. Intel. Robots and Systems (IROS)*, 2016.

[24] Carl Edward Rasmussen and Christopher KI Williams. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, 2006.

[25] Paul Rolland, Jonathan Scarlett, Ilija Bogunovic, and Volkan Cevher. High-dimensional Bayesian optimization via additive models with overlapping groups. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 298–307, 2018.

[26] Binxin Ru, Michael Osborne, and Mark McLeod. Fast information-theoretic Bayesian optimisation. *arXiv preprint arXiv:1711.00673*, 2017.

[27] Jonathan Scarlett, Ilijia Bogunovic, and Volkan Cevher. Lower bounds on regret for noisy Gaussian process bandit optimization. In *Conference on Learning Theory (COLT)*, 2017.

[28] Shubhanshu Shekhar and Tara Javidi. Gaussian process bandits with adaptive discretization. *arXiv preprint arXiv:1712.01447*, 2017.

[29] Aman Sinha, Hongseok Namkoong, and John Duchi. Certifiable distributional robustness with principled adversarial training. In *International Conference on Learning Representations (ICLR)*, 2018.

[30] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural information Processing Systems (NIPS)*, pages 2951–2959, 2012.

[31] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning (ICML)*, pages 1015–1022, 2010.

[32] Matthew Staib, Bryan Wilder, and Stefanie Jegelka. Distributionally robust submodular maximization. *arXiv preprint arXiv:1802.05249*, 2018.

[33] Yanan Sui, Alkis Gotovos, Joel Burdick, and Andreas Krause. Safe exploration for optimization with Gaussian processes. In *International Conference on Machine Learning (ICML)*, pages 997–1005, 2015.

[34] Hastagiri P Vanchinathan, Isidor Nikolic, Fabio De Bona, and Andreas Krause. Explore-exploit in top-n recommender systems via Gaussian processes. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 225–232. ACM, 2014.

[35] Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient Bayesian optimization. In *International Conference on Machine Learning (ICML)*, pages 3627–3635, 2017.

[36] Zi Wang, Chengtao Li, Stefanie Jegelka, and Pushmeet Kohli. Batched high-dimensional Bayesian optimization via structural kernel learning. In *International Conference on Machine Learning (ICML)*, pages 3656–3664, 2017.

[37] Bryan Wilder. Equilibrium computation for zero sum games with submodular structure. In *Conference on Artificial Intelligence (AAAI)*, 2017.

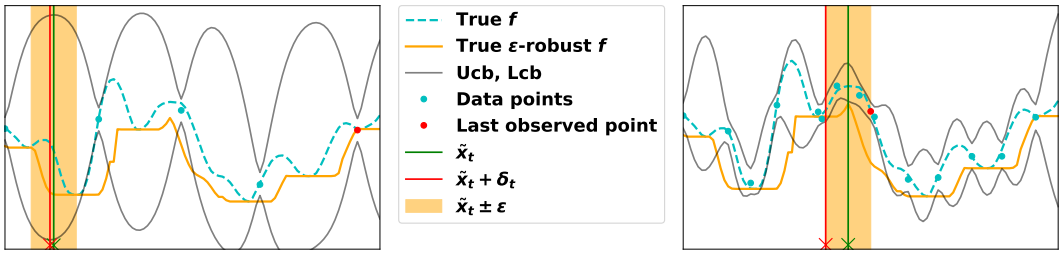
Supplementary Material

Adversarially Robust Optimization with Gaussian Processes

Ilija Bogunovic, Jonathan Scarlett, Stefanie Jegelka and Volkan Cevher (NeurIPS 2018)

A Illustration of StableOpt's Execution

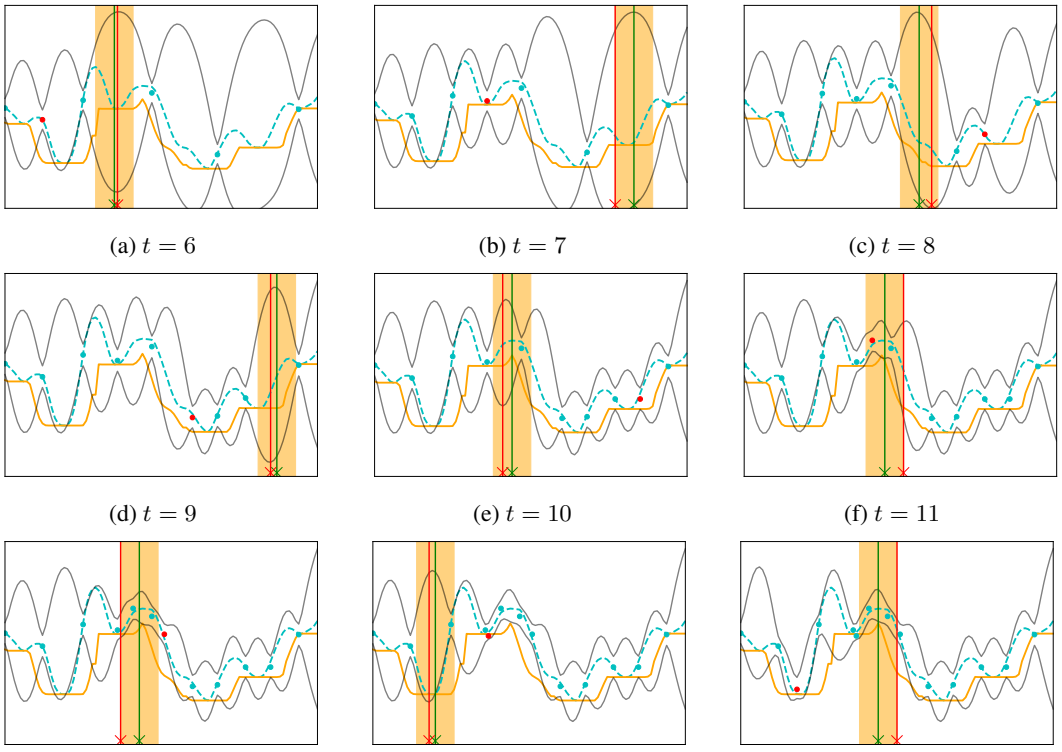
The following figure gives an example of the selection procedure of StableOpt at two different time steps:



(a) $t = 5$ (b) $t = 15$

Figure 4: An execution of StableOpt on the running example. We observe that after $t = 15$ steps, $\tilde{\mathbf{x}}_t$ obtained in Eq. 13 corresponds to $\mathbf{x}^*_\epsilon$.

The intermediate time steps are illustrated as follows:



(a) $t = 6$ (b) $t = 7$ (c) $t = 8$

(d) $t = 9$ (e) $t = 10$ (f) $t = 11$

(g) $t = 12$ (h) $t = 13$ (i) $t = 14$

B Proofs of Theoretical Results

B.1 Proof of Theorem 1 (upper bound)

Recall that $\tilde{\mathbf{x}}_t$ is the point computed by StableOpt in (13) at time t, and that $\boldsymbol{\delta}_t$ corresponds to the perturbation obtained in StableOpt (Line 3) at time t. In the following, we condition on the event in Lemma 1 holding true, meaning that ucb_t and lcb_t provide valid confidence bounds as per (15). As stated in the lemma, this holds with probability at least $1-\xi$.

By the definition of ϵ-instant regret, we have

$$
\begin{aligned}
r_\epsilon(\tilde{\mathbf{x}}_t) &= \max_{\mathbf{x}\in D} \min_{\boldsymbol{\delta}\in\Delta_\epsilon(\mathbf{x})} f(\mathbf{x}+\boldsymbol{\delta}) - \min_{\boldsymbol{\delta}\in\Delta_\epsilon(x_t)} f(\tilde{\mathbf{x}}_t+\boldsymbol{\delta}) && (32)\\
&\le \max_{\mathbf{x}\in D} \min_{\boldsymbol{\delta}\in\Delta_\epsilon(\mathbf{x})} f(\mathbf{x}+\boldsymbol{\delta}) - \min_{\boldsymbol{\delta}\in\Delta_\epsilon(\tilde{\mathbf{x}}_t)} \mathrm{lcb}_{t-1}(\tilde{\mathbf{x}}_t+\boldsymbol{\delta}) && (33)\\
&= \max_{\mathbf{x}\in D} \min_{\boldsymbol{\delta}\in\Delta_\epsilon(\mathbf{x})} f(\mathbf{x}+\boldsymbol{\delta}) - \mathrm{lcb}_{t-1}(\tilde{\mathbf{x}}_t+\boldsymbol{\delta}_t) && (34)\\
&\le \max_{\mathbf{x}\in D} \min_{\boldsymbol{\delta}\in\Delta_\epsilon(\mathbf{x})} \mathrm{ucb}_{t-1}(\mathbf{x}+\boldsymbol{\delta}) - \mathrm{lcb}_{t-1}(\tilde{\mathbf{x}}_t+\boldsymbol{\delta}_t) && (35)\\
&= \min_{\boldsymbol{\delta}\in\Delta_\epsilon(\tilde{\mathbf{x}}_t)} \mathrm{ucb}_{t-1}(\tilde{\mathbf{x}}_t+\boldsymbol{\delta}) - \mathrm{lcb}_{t-1}(\tilde{\mathbf{x}}_t+\boldsymbol{\delta}_t) && (36)\\
&\le \mathrm{ucb}_{t-1}(\tilde{\mathbf{x}}_t+\boldsymbol{\delta}_t) - \mathrm{lcb}_{t-1}(\tilde{\mathbf{x}}_t+\boldsymbol{\delta}_t) && (37)\\
&= 2\beta_t^{1/2}\sigma_{t-1}(\tilde{\mathbf{x}}_t+\boldsymbol{\delta}_t), && (38)
\end{aligned}
$$

where (33) and (35) follow from Lemma 1, (34) follows since $\boldsymbol{\delta}_t$ minimizes lcb_{t-1} by definition, (36) follows since $\tilde{\mathbf{x}}_t$ maximizes the robust upper confidence bound by definition, (37) follows by upper bounding the minimum by the specific choice $\boldsymbol{\delta}_t \in \Delta_\epsilon(\mathbf{x}_t)$, and (38) follows since the upper and lower confidence bounds are separated by $2\beta_t^{1/2}\sigma_{t-1}(\cdot)$ according to their definitions in (12).

In fact, the analysis from (33) to (38) shows that the following *pessimistic estimate* of $r_\epsilon(\tilde{\mathbf{x}}_t)$ is upper bounded by $2\beta_t^{1/2}\sigma_{t-1}(\tilde{\mathbf{x}}_t+\boldsymbol{\delta}_t)$:

$$
\overline{r}_\epsilon(\tilde{\mathbf{x}}_t) = \max_{\mathbf{x}\in D} \min_{\boldsymbol{\delta}\in\Delta_\epsilon(\mathbf{x})} f(\mathbf{x}+\boldsymbol{\delta}) - \min_{\boldsymbol{\delta}\in\Delta_\epsilon(\tilde{\mathbf{x}}_t)} \mathrm{lcb}_{t-1}(\tilde{\mathbf{x}}_t+\boldsymbol{\delta}). \tag{39}
$$

Unlike $r_\epsilon(\tilde{\mathbf{x}}_t)$, the algorithm has the required knowledge to identify the value of $t \in \{1,\dots,T\}$ with the smallest $\overline{r}_\epsilon(\tilde{\mathbf{x}}_t)$. Specifically, the first term on the right-hand side of (39) does not depend on t, so the smallest $\overline{r}_\epsilon(\tilde{\mathbf{x}}_t)$ is achieved by $\mathbf{x}^{(T)}$ defined in (17). Since the minimum is upper bounded by the average, it follows that

$$
\begin{aligned}
r_\epsilon(\mathbf{x}^{(T)}) &\le \overline{r}_\epsilon(\mathbf{x}^{(T)}) && (40)\\
&\le \frac{1}{T}\sum_{t=1}^{T} 2\beta_t^{1/2}\sigma_{t-1}(\tilde{\mathbf{x}}_t+\boldsymbol{\delta}_t) && (41)\\
&\le \frac{2\beta_T^{1/2}}{T}\sum_{t=1}^{T}\sigma_{t-1}(\tilde{\mathbf{x}}_t+\boldsymbol{\delta}_t), && (42)
\end{aligned}
$$

where (41) uses (38), and (42) uses the monotonicity of β_T. Next, we claim that

$$
2\sum_{t=1}^{T}\sigma_{t-1}(\tilde{\mathbf{x}}_t+\boldsymbol{\delta}_t) \le \sqrt{C_1 T\gamma_T}, \tag{43}
$$

where $C_1 = 8/\log(1+\sigma^{-2})$. In fact, this is a special case of the well-known result [31, Lemma 5.4],[4] which upper bounds the sum of posterior standard deviations of sampled points in terms of the information gain γ_T (recall that StableOpt samples at location $\tilde{\mathbf{x}}_t+\boldsymbol{\delta}_t$). Combining (42)–(43) and re-arranging, we deduce that after T satisfies $\frac{T}{\beta_T\gamma_T} \ge \frac{C_1}{\eta^2}$, the ϵ-instant regret is at most η, thus completing the proof.

[4] More precisely, [31, Lemma 5.4] alongside an application of the Cauchy-Schwarz inequality as in [31].

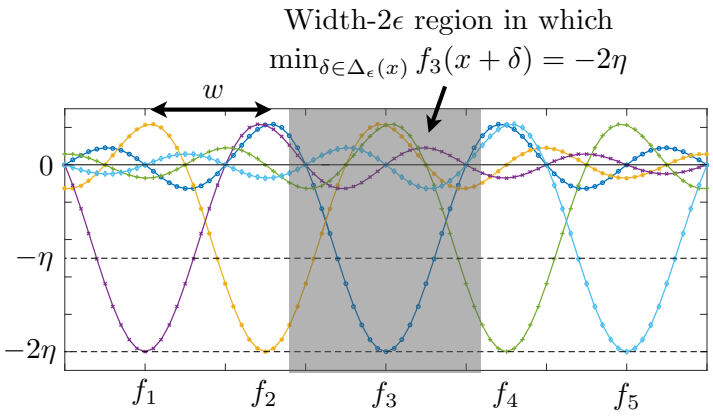


Figure 6: Illustration of functions $f_1, \ldots, f_5$ equal to a common function shifted by various multiples of a given parameter w. In the ϵ-stable setting, there is a wide region (shown in gray for the dark blue curve f_3) within which the perturbed function value equals -2η.

B.2 Proof of Theorem 2 (lower bound)

Our lower bounding analysis builds heavily on that of the non-robust optimization setting with $f \in \mathcal{F}_k(B)$ studied in [27], but with important differences. Roughly speaking, the analysis of [27] is based on the difficulty of finding a very narrow "bump" of height 2η in a function whose values are mostly close to zero. In the ϵ-stable setting, however, even the points around such a bump will be adversarially perturbed to another point whose function value is nearly zero. Hence, all points are essentially equally bad.

To overcome this challenge, we consider the reverse scenario: Most of the function values are still nearly zero, but there exists a narrow *valley* of depth -2η. This means that every point within an ϵ-ball around the function minimizer will be perturbed to the point with value -2η. Hence, a constant fraction of the volume is still 2η-suboptimal, and it is impossible to avoid this region with high probability unless the time horizon T is sufficiently large. An illustration is given in Figure 6, with further details below.

We now proceed with the formal proof.

B.2.1 Preliminaries

Recall that we are considering an arbitrary given (deterministic) GP optimization algorithm. More precisely, such an algorithm consists of a sequence of decision functions that return a sampling location $\mathbf{x}_t$ based on $y_1, \ldots, y_{t-1}$, and an additional decision function that reports the final point $\mathbf{x}^{(T)}$ based on $y_1, \ldots, y_T$. The points $\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}$ (or $\mathbf{x}_1, \ldots, \mathbf{x}_T$) do not need to be treated as additional inputs to these functions, since $(\mathbf{x}_1, \ldots, \mathbf{x}_{t-1})$ is a deterministic function of $(y_1, \ldots, y_{t-1})$.

We first review several useful results and techniques from [27]:

- We lower bound the worst-case ϵ-regret within $\mathcal{F}_k(B)$ by the ϵ-regret averaged over a suitably-designed finite collection $\{f_1, \ldots, f_M\} \subset \mathcal{F}_k(B)$ of size M.
- We choose each $f_m(\mathbf{x})$ to be a shifted version of a common function $g(\mathbf{x})$ on $\mathbb{R}^p$. Specifically, each $f_m(\mathbf{x})$ is obtained by shifting $g(\mathbf{x})$ by a different amount, and then cropping to $D = [0,1]^p$. For our purposes, we require $g(\mathbf{x})$ to satisfy the following properties:
  1. The RKHS norm in $\mathbb{R}^p$ is bounded, $\|g\|_k \le B$;
  2. We have (i) $g(\mathbf{x}) \in [-2\eta, 2\eta]$ with minimum value $g(0) = -2\eta$, and (ii) there is a "width" w such that $g(\mathbf{x}) > -\eta$ for all $\|\mathbf{x}\|_\infty \ge w$;
  3. There are absolute constants $h_0 > 0$ and $\zeta > 0$ such that $g(\mathbf{x}) = \frac{2\eta}{h_0} h\big(\frac{\mathbf{x}\zeta}{w}\big)$ for some function $h(\mathbf{z})$ that decays faster than any finite power of $\|\mathbf{z}\|_2^{-1}$ as $\|\mathbf{z}\|_2 \to \infty$.

Letting $g(\mathbf{x})$ be such a function, we construct the M functions by shifting $g(\mathbf{x})$ so that each $f_m(\mathbf{x})$ is centered on a unique point in a uniform grid, with points separated by w in each dimension. Since $D = [0,1]^p$, one can construct

$$M = \left\lfloor \left(\frac{1}{w}\right)^p \right\rfloor \tag{44}$$

such functions. We will use this construction with $w \ll 1$, so that there is no risk of having $M = 0$, and in fact M can be assumed larger than any desired absolute constant.

- It is shown in [27] that the above properties[5] can be achieved with

$$M = \left\lfloor \left(\frac{r\sqrt{\log \frac{B(2\pi l^2)^{p/4}h(0)}{2\eta}}}{\zeta \pi l} \right)^p \right\rfloor \tag{45}$$

in the case of the SE kernel, and with

$$M = \left\lfloor \left(\frac{Bc_3}{\eta}\right)^{p/\nu} \right\rfloor \tag{46}$$

in the case of the Matérn kernel, where

$$c_3 := \left(\frac{r}{\zeta}\right)^{\nu} \cdot \left(\frac{c_2^{-1/2}}{2(8\pi^2)^{(\nu+p/2)/2}} \right), \tag{47}$$

and where $c_2 > 0$ is an absolute constant. Note that these values of M amount to choosing w in (44), and the assumption of sufficiently small $\frac{\eta}{B}$ in the theorem statement ensures that $M \gg 1$ (or equivalently $w \ll 1$) as stated above.

- Property 2 above ensures that the "robust" function value $\min_{\boldsymbol{\delta} \in \Delta_\epsilon(\mathbf{x})} f(\mathbf{x})$ equals -2η for any $\mathbf{x}$ whose ϵ-neighborhood includes the minimizer $\mathbf{x}_{\min}$ of f, while being $-\eta$ or higher for any input whose entire ϵ-neighborhood is separated from $\mathbf{x}_{\min}$ by at least w. For $w \ll 1$ and $\epsilon < 0.5$, a point of the latter type is guaranteed to exist, which implies

$$r_\epsilon(\mathbf{x}) \geq \eta \tag{48}$$

for any $\mathbf{x}$ whose ϵ-neighborhood includes $\mathbf{x}_{\min}$.

In addition, we introduce the following notation, also used in [27]:

- The probability density function of the output sequence $\mathbf{y} = (y_1, \ldots, y_T)$ when the underlying function is f_m is denoted by $P_m(\mathbf{y})$. We also define $f_0(\mathbf{x}) = 0$ to be the zero function, and define $P_0(\mathbf{y})$ analogously for the case that the optimization algorithm is run on f_0. Expectations and probabilities (with respect to the noisy observations) are similarly written as $\mathbb{E}_m$, $\mathbb{P}_m$, $\mathbb{E}_0$, and $\mathbb{P}_0$ when the underlying function is f_m or f_0. On the other hand, in the absence of a subscript, $\mathbb{E}[\cdot]$ and $\mathbb{P}[\cdot]$ are taken with respect to the noisy observations *and* the random function f drawn uniformly from $\{f_1, \ldots, f_M\}$ (recall that we are lower bounding the worst case by this average).
- Let $\{\mathcal{R}_m\}_{m=1}^M$ be a partition of the domain into M regions according the above-mentioned uniform grid, with f_m taking its minimum value of -2η in the centre of $\mathcal{R}_m$. Moreover, let j_t be the index at time t such that $\mathbf{x}_t$ falls into $\mathcal{R}_{j_t}$; this can be thought of as a quantization of $\mathbf{x}_t$.
- Define the maximum (absolute) function value within a given region $\mathcal{R}_j$ as

$$\overline{v}_m^j := \max_{\mathbf{x} \in \mathcal{R}_j} |f_m(\mathbf{x})|, \tag{49}$$

and the maximum KL divergence to P_0 within the region as

$$\overline{D}_m^j := \max_{\mathbf{x} \in \mathcal{R}_j} D(P_0(\cdot|\mathbf{x}) \| P_m(\cdot|\mathbf{x})), \tag{50}$$

where $P_m(y|\mathbf{x})$ is the distribution of an observation y for a given selected point $\mathbf{x}$ under the function f_m, and similarly for $P_0(y|\mathbf{x})$.

[5] Here $g(\mathbf{x})$ plays the role of $-g(\mathbf{x})$ in [27] due to the discussion at the start of this appendix, but otherwise the construction is identical.

- Let $N_j \in \{0, \ldots, T\}$ be a random variable representing the number of points from $\mathcal{R}_j$ that are selected throughout the T rounds.

Next, we present several useful lemmas. The following well-known change-of-measure result, which can be viewed as a form of Le Cam's method, has been used extensively in both discrete and continuous bandit problems.

Lemma 2. [1, p. 27] *For any function $a(\mathbf{y})$ taking values in a bounded range $[0, A]$, we have*

$$\big|\mathbb{E}_m[a(\mathbf{y})] - \mathbb{E}_0[a(\mathbf{y})]\big| \le A\, d_{\mathrm{TV}}(P_0, P_m) \tag{51}$$

$$\le A\, \sqrt{D(P_0 \| P_m)}, \tag{52}$$

where $d_{\mathrm{TV}}(P_0, P_m) = \frac{1}{2}\int_{\mathbb{R}^T} |P_0(\mathbf{y}) - P_m(\mathbf{y})|\, d\mathbf{y}$ is the total variation distance.

We briefly remark on some slight differences here compared to [1, p. 27]. There, only $\mathbb{E}_m[a(\mathbf{y})] - \mathbb{E}_0[a(\mathbf{y})]$ is upper bounded in terms of $d_{\mathrm{TV}}(P_0, P_m)$, but one easily obtains the same upper bound on $\mathbb{E}_0[a(\mathbf{y})] - \mathbb{E}_m[a(\mathbf{y})]$ by interchanging the roles of P_0 and P_m. The step (52) follows from Pinsker's inequality, $d_{\mathrm{TV}}(P_0, P_m) \le \sqrt{\frac{D(P_0\|P_m)}{2}}$, and by upper bounding $\frac{1}{\sqrt{2}} \le 1$ to ease the notation.

The following result simplifies the divergence term in (52).

Lemma 3. [27, Eq. (44)] *Under the preceding definitions, we have*

$$D(P_0 \| P_m) \le \sum_{j=1}^{M} \mathbb{E}_0[N_j]\, \overline{D}_m^j. \tag{53}$$

The following well-known property gives a formula for the KL divergence between two Gaussians.

Lemma 4. [27, Eq. (36)] *For P_1 and P_2 being Gaussian with means (μ_1, μ_2) and a common variance σ^2, we have*

$$D(P_1 \| P_2) = \frac{(\mu_1 - \mu_2)^2}{2\sigma^2}. \tag{54}$$

Finally, we have the following technical result regarding the "needle-in-haystack" type function constructed above.

Lemma 5. [27, Lemma 7] *The functions $\{f_m\}_{m=1}^M$ corresponding to* (45)–(46) *are such that the quantities $\overline{v}_m^j$ satisfy $\sum_{m=1}^M (\overline{v}_m^j)^2 = O(\eta^2)$ for all j.*

B.2.2 Analysis of the average ϵ-stable regret

Let $J_{\mathrm{bad}}(m)$ be the set of j such that all $\mathbf{x} \in \mathcal{R}_j$ yield $\min_{\boldsymbol{\delta} \in \Delta_\epsilon(\mathbf{x})} f(\mathbf{x} + \boldsymbol{\delta}) = -2\eta$ when the true function is f_m, and define $\mathcal{R}_{\mathrm{bad}}(m) = \cup_{j \in J_{\mathrm{bad}}(m)} \mathcal{R}_j$. By the ϵ-regret lower bound in (48), we have

$$\mathbb{E}_m[r_\epsilon(\mathbf{x}^{(T)})] \ge \eta \mathbb{P}_m[\mathbf{x}^{(T)} \in \mathcal{R}_{\mathrm{bad}}(m)] \tag{55}$$

$$\ge \eta\left(\mathbb{P}_0[\mathbf{x}^{(T)} \in \mathcal{R}_{\mathrm{bad}}(m)] - \sqrt{D(P_0\|P_m)}\right) \tag{56}$$

$$\ge \eta\left(\mathbb{P}_0[\mathbf{x}^{(T)} \in \mathcal{R}_{\mathrm{bad}}(m)] - \sqrt{\sum_{j=1}^{M} \mathbb{E}_0[N_j]\, \overline{D}_m^j}\right), \tag{57}$$

where (56) follows from Lemma 2 with $a(\mathbf{y}) = \mathbf{1}\{\mathbf{x}^{(T)} \in \mathcal{R}_{\mathrm{bad}}(m)\}$ and $A = 1$ (recall that $\mathbf{x}^{(T)}$ is a function of $\mathbf{y} = (y_1, \ldots, y_T)$), and (57) follows from Lemma 3. Averaging over m uniform on $\{1, \ldots, M\}$, we obtain

$$\mathbb{E}[r_\epsilon(\mathbf{x}^{(T)})] \ge \eta\left(\frac{1}{M}\sum_{m=1}^{M} \mathbb{P}_0[\mathbf{x}^{(T)} \in \mathcal{R}_{\mathrm{bad}}(m)] - \frac{1}{M}\sum_{m=1}^{M}\sqrt{\sum_{j=1}^{M} \mathbb{E}_0[N_j]\, \overline{D}_m^j}\right). \tag{58}$$

We proceed by bounding the two terms separately.

- We first claim that

$$\frac{1}{M}\sum_{m=1}^{M}\mathbb{P}_0[\mathbf{x}^{(T)}\in\mathcal{R}_{\mathrm{bad}}(m)]\geq C_1 \tag{59}$$

for some $C_1>0$. To show this, it suffices to prove that any given $\mathbf{x}^{(T)}\in D$ is in at least a constant fraction of the $\mathcal{R}_{\mathrm{bad}}(m)$ regions, of which there are M. This follows from the fact that the ϵ-ball centered at $\mathbf{x}_{m,\min}=\arg\min_{\mathbf{x}\in D} f_m(\mathbf{x})$ takes up a constant fraction of the volume of D, where the constant depends on both the stability parameter ϵ and the dimension p. A small caveat is that because the definition of $\mathcal{R}_{\mathrm{bad}}$ insists that the *every* point in the region $\mathcal{R}_j$ is within distance ϵ of $\mathbf{x}_{m,\min}$, the left-hand side of (59) may be slightly below the relevant ratio of volumes above. However, since Theorem 2 assumes that $\frac{\eta}{B}$ is sufficiently small, the choices of M in (45) and (46) ensure that M is sufficiently large for this "quantization" effect to be negligible.

- For the second term in (58), we claim that

$$\frac{1}{M}\sum_{m=1}^{M}\sqrt{\sum_{j=1}^{M}\mathbb{E}_0[N_j]\overline{D}_m^j}\leq C_2\frac{\eta}{\sigma}\sqrt{\frac{T}{M}} \tag{60}$$

for some $C_2>0$. To see this, we write

$$\begin{aligned}
&\frac{1}{M}\sum_{m=1}^{M}\sqrt{\sum_{j=1}^{M}\mathbb{E}_0[N_j]\overline{D}_m^j}\\
&\quad=O\Big(\frac{1}{\sigma}\Big)\cdot\frac{1}{M}\sum_{m=1}^{M}\sqrt{\sum_{j=1}^{M}\mathbb{E}_0[N_j](\overline{v}_m^j)^2} &(61)\\
&\quad\leq O\Big(\frac{1}{\sigma}\Big)\cdot\sqrt{\frac{1}{M}\sum_{m=1}^{M}\sum_{j=1}^{M}\mathbb{E}_0[N_j](\overline{v}_m^j)^2} &(62)\\
&\quad=O\Big(\frac{1}{\sigma}\Big)\cdot\sqrt{\frac{1}{M}\sum_{j=1}^{M}\mathbb{E}_0[N_j]\Big(\sum_{m=1}^{M}(\overline{v}_m^j)^2\Big)} &(63)\\
&\quad=O\Big(\frac{\eta}{\sqrt{M}\sigma}\Big)\cdot\sqrt{\sum_{j=1}^{M}\mathbb{E}_0[N_j]} &(64)\\
&\quad=O\Big(\frac{\sqrt{T}\eta}{\sqrt{M}\sigma}\Big), &(65)
\end{aligned}$$

where (61) follows since the divergence $D(P_0(\cdot|\mathbf{x})\|P_m(\cdot|\mathbf{x}))$ associated with a point $\mathbf{x}$ having value $v(\mathbf{x})$ is $\frac{v(\mathbf{x})^2}{2\sigma^2}$ (*cf.*, (54)), (62) follows from Jensen's inequality, (64) follows from Lemma 5, and (65) follows from $\sum_j N_j=T$.

Substituting (59) and (60) into (58), we obtain

$$\mathbb{E}[r_\epsilon(\mathbf{x}^{(T)})]\geq\eta\Big(C_1-C_2\frac{\eta}{\sigma}\sqrt{\frac{T}{M}}\Big), \tag{66}$$

which implies that the regret is lower bounded by $\Omega(\eta)$ unless $T=\Omega\big(\frac{M\sigma^2}{\eta^2}\big)$. Substituting M from (45) and (46), we deduce that the conditions on T in the theorem statement are necessary to achieve average regret $\mathbb{E}[r_\epsilon(\mathbf{x}^{(T)})]=O(\eta)$ with a sufficiently small implied constant.

B.2.3 From average to high-probability regret

Recall that we are considering functions whose values lie in the range $[-2\eta,2\eta]$, implying that $r_\epsilon(\mathbf{x}^{(T)})\leq 4\eta$. Letting T_η be the lower bound on T derived above for achieving average regret

$O(\eta)$ (i.e., we have $\mathbb{E}[r_\epsilon^{(T_\eta)}] = \Omega(\eta)$), it follows from the reverse Markov inequality (i.e., Markov's inequality applied to the random variable $4\eta - r_\epsilon^{(T_\eta)}$) that

$$\mathbb{P}[r_\epsilon(\mathbf{x}^{(T_\eta)}) \geq c\eta] \geq \frac{\Omega(\eta) - c\eta}{4\eta - c\eta} \tag{67}$$

for any $c > 0$ sufficiently small for the numerator and denominator to be positive. The right-hand side is lower bounded by a constant for any such c, implying that the probability of achieving ϵ-regret at most $c\eta$ cannot be arbitrarily close to one. By renaming $c\eta$ as η', it follows that in order to achieve some target ϵ-stable regret η' with probability sufficiently close to one, a lower bound of the same form as the average regret bound holds. In other words, the conditions on T in the theorem statement remain necessary also for the high-probability regret.

We emphasize that Theorem 2 concerns the high-probability regret when "high probability" means *sufficiently close to one* as a function of ϵ, p, and the kernel parameters (but still constant with respect to T and η). We do not claim a lower bound under any particular *given* success probability (e.g., η-optimality with probability at least $\frac{3}{4}$).

C Details on Variations from Section 4

We claim that the StableOpt variations and theoretical results outlined in Section 4 are in fact special cases of Algorithm 1 and Theorem 1, despite being seemingly quite different. The idea behind this claim is that Algorithm 1 and Theorem 1 allow for the "distance" function $d(\cdot, \cdot)$ to be completely arbitrary, so we may choose it in rather creative/unconventional ways.

In more detail, we have the following:

- For the unknown parameter setting $\max_{\mathbf{x} \in D} \min_{\boldsymbol{\theta} \in \Theta} f(\mathbf{x}, \boldsymbol{\theta})$, we replace $\mathbf{x}$ in the original setting by the concatenated input $(\mathbf{x}, \boldsymbol{\theta})$, and set

  $$d((\mathbf{x}, \boldsymbol{\theta}), (\mathbf{x}', \boldsymbol{\theta}')) = \|\mathbf{x} - \mathbf{x}'\|_2. \tag{68}$$

  If we then set $\epsilon = 0$, we find that the input $\mathbf{x}$ experiences no perturbation, whereas $\boldsymbol{\theta}$ may be perturbed arbitrarily, thereby reducing (7) to $\max_{\mathbf{x} \in D} \min_{\boldsymbol{\theta} \in \Theta} f(\mathbf{x}, \boldsymbol{\theta})$ as desired.

- For the robust estimation setting, we again use the concatenated input $(\mathbf{x}, \boldsymbol{\theta})$. To avoid overloading notation, we let $d_0(\boldsymbol{\theta}, \boldsymbol{\theta}')$ denote the distance function (applied to $\boldsymbol{\theta}$ alone) adopted for this case in Section 4. We set

  $$d((\mathbf{x}, \boldsymbol{\theta}), (\mathbf{x}', \boldsymbol{\theta}')) = \begin{cases} d_0(\boldsymbol{\theta}, \boldsymbol{\theta}') & \mathbf{x} = \mathbf{x}' \\ \infty & \mathbf{x} \neq \mathbf{x}'. \end{cases} \tag{69}$$

  Due to the second case, the input $\mathbf{x}$ experiences no perturbation, since doing so would violate the distance constraint of ϵ. We are then left with $\mathbf{x} = \mathbf{x}'$ and $d_0(\boldsymbol{\theta}, \boldsymbol{\theta}') \leq \epsilon$, as required.

- For the grouped setting $\max_{G \in \mathcal{G}} \min_{\mathbf{x} \in G} f(\mathbf{x})$, we adopt the function

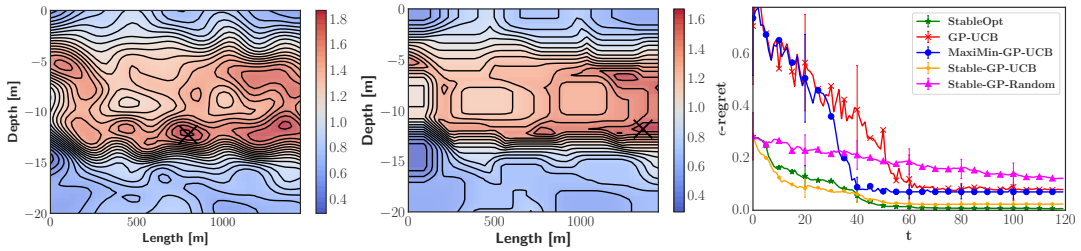
  $$d(\mathbf{x}, \mathbf{x}') = \mathbf{1}\{\mathbf{x} \text{ and } \mathbf{x}' \text{ are in different groups}\}, \tag{70}$$

  and set $\epsilon = 0$. Considering the formulation in (7), we find that any two inputs $\mathbf{x}$ and $\mathbf{x}'$ yield the same ϵ-stable objective function, and hence, reporting a point $\mathbf{x}$ is equivalent to reporting its group G. As a result, (7) reduces to the desired formulation $\max_{G \in \mathcal{G}} \min_{\mathbf{x} \in G} f(\mathbf{x})$.

The variations of StableOpt described in (20)–(26), as well as the corresponding theoretical results outlined in Section 4, follow immediately by substituting the respective choices of $d(\cdot, \cdot)$ and ϵ above into Algorithm 1 and Theorem 1. It should be noted that in the first two examples, the definition of γ_t in (14) is modified to take the maximum over not only $\mathbf{x}_1, \cdots, \mathbf{x}_t$, but also $\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_t$.

D Lake Data Experiment

We consider an application regarding environmental monitoring of inland waters, using a data set containing 2024 in situ measurements of chlorophyll concentration within a vertical transect plane, collected by an autonomous surface vessel in Lake Zürich. This data set was considered in previous



(a) Chlorophyll concentration (b) Robust objective (c) ϵ-regret

Figure 7: Experiment on the Zürich lake dataset; In the later rounds StableOpt is the only method that reports a near-optimal ϵ-stable point.

works such as [7, 15] to detect regions of high concentration. In these works, the goal was to locate all regions whose concentration exceeds a pre-defined threshold.

Here we consider a different goal: We seek to locate a region of a given size such that the concentration throughout the region is as high as possible (in the max-min sense). This is of interest in cases where high concentration only becomes relevant when it is spread across a sufficiently wide area. We consider rectangular regions with different pre-specified lengths in each dimension:

$$\Delta_{\epsilon_D,\epsilon_L}(\mathbf{x}) = \{\mathbf{x}' - \mathbf{x} : \mathbf{x}' \in D,\ |x_D - x'_D| \leq \epsilon_D \cap |x_L - x'_L| \leq \epsilon_L\}, \tag{71}$$

where $\mathbf{x} = (x_D, x_L)$ and $\mathbf{x}' = (x'_D, x'_L)$ indicate the depth and length, and we denote the corresponding stability parameters by (ϵ_D, ϵ_L). This corresponds to $d(\cdot, \cdot)$ being a weighted ℓ_∞-norm.

We evaluate each algorithm on a 50×50 grid of points, with the corresponding values coming from the GP posterior that was derived using the original data. We use the Matérn-5/2 ARD kernel, setting its hyperparameters by maximizing the likelihood on a second (smaller) available dataset. The parameters ϵ_D and ϵ_L are set to 1.0 and 100.0, respectively. The stability requirement changes the global maximum and its location, as can be observed in Figure 7. The number of sampling rounds is $T = 120$, and each algorithm is initialized with the same 10 random data points and corresponding observations. The performance is averaged over 100 different runs, where every run corresponds to a different random initialization. In this experiment, Stable-GP-UCB achieves the smallest ϵ-regret in the early rounds, while in the later rounds StableOpt is the only method that reports a near-optimal ϵ-stable point.