Corruption-Tolerant Gaussian Process Bandit Optimization

Ilija Bogunovic
ETH Zürich

Andreas Krause
ETH Zürich

Jonathan Scarlett
National University of Singapore

Abstract

We consider the problem of optimizing an unknown (typically non-convex) function with a bounded norm in some Reproducing Kernel Hilbert Space (RKHS), based on noisy bandit feedback. We consider a novel variant of this problem in which the point evaluations are not only corrupted by random noise, but also adversarial corruptions. We introduce an algorithm Fast-Slow GP-UCB based on Gaussian process methods, randomized selection between two instances labeled "fast" (but non-robust) and "slow" (but robust), enlarged confidence bounds, and the principle of optimism under uncertainty. We present a novel theoretical analysis upper bounding the cumulative regret in terms of the corruption level, the time horizon, and the underlying kernel, and we argue that certain dependencies cannot be improved. We observe that distinct algorithmic ideas are required depending on whether one is required to perform well in both the corrupted and non-corrupted settings, and whether the corruption level is known or not.

1 Introduction

Bandit optimization problems on large or continuous domains have far-reaching applications in modern machine learning and data science, including robotics [Lizotte et al., 2007], hyperparameter tuning [Snoek et al., 2012], recommender systems [Vanchinathan et al., 2014], environmental monitoring [Srinivas et al., 2010], and more. To make such problems tractable, one needs to exploit correlations between the rewards of "similar" actions. In the *kernelized multi-armed bandit* (MAB) problem, this is done by utilizing smoothness in the form of a low function norm in some Reproducing Kernel Hilbert Space (RKHS), permitting the application of Gaussian process (GP) methods [Srinivas et al., 2010, Chowdhury and Gopalan, 2017]. See [Rasmussen and Williams, 2006, Ch. 6] for an introduction to the connections between GPs and RKHS functions.

Key theoretical developments for the RKHS optimization problem have included both upper and lower bounds on the performance, measured via some notion of regret [Srinivas et al., 2010, Chowdhury and Gopalan, 2017, Scarlett et al., 2017]. The vast majority of these results have focused only on zero-mean additive noise in the point evaluations, and as a result, it is unclear to what extent the performance degrades under *adversarial corruptions*. Such considerations are of significant interest under erratic or unpredictable sources of corruption, and particularly arise when the samples may be perturbed by a malicious adversary. As we argue in Section 2, prominent algorithms such as GP-UCB [Srinivas et al., 2010] can be quite brittle in the face of such corruptions.

In this paper, we study the optimization of RKHS functions with both random noise and adversarial corruptions. We propose a novel algorithm and regret analysis building on recently-proposed techniques for the finite-arm stochastic MAB setting [Lykouris et al., 2018]. Specifically, we present a randomized algorithm Fast-Slow GP-UCB based on randomly choosing between a "fast" non-robust instance, and a "slow" robust instance. We bound the cumulative regret of Fast-Slow GP-UCB in terms of the adversarial corruption level, time horizon, and underlying kernel.

The kernelized setting comes with highly non-trivial additional challenges compared to the finite-arm setting, primarily due to the infinite action space and correlations between their associated function values. In particular, while correlations are undoubtedly beneficial in the non-corrupted setting (taking a given action permits learning something about similar actions), this benefit can lead to a hindrance in the corrupted setting: An adversary that corrupts a given sample can potentially *damage* our belief regarding *many* nearby function values. Moving beyond independent arms was posed as a open problem in [Gupta et al., 2019, Sec. 5.3].

Related work on GP optimization. Numerous

Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108. Copyright 2020 by the author(s).

GP-based bandit optimization algorithms have been proposed in recent years [Srinivas et al., 2010, Hennig and Schuler, 2012, Hernández-Lobato et al., 2014, Bogunovic et al., 2016b, Wang and Jegelka, 2017, Shekhar and Javidi, 2018, Ru et al., 2017]. Beyond the standard setting, several important extensions have been considered, including multi-fidelity [Bogunovic et al., 2016b, Kandasamy et al., 2017, Song et al., 2019], contextual and time-varying settings [Krause and Ong, 2011, Valko et al., 2013, Bogunovic et al., 2016a], safety requirements [Sui et al., 2015], high-dimensional settings [Djolonga et al., 2013, Kandasamy et al., 2015, Rolland et al., 2018], and many more.

Certain types of corruption-tolerant GP-based optimization algorithms have been explored previously, with the defining features including (i) whether the corruption applies to the *input* (i.e., action) or the *output* (i.e., reward function), (ii) whether *all samples* are corrupted, or only a *final reported point* is corrupted, and (iii) whether the corruptions are random or adversarial. The case of random input noise on all samples was studied in [Beland and Nair, 2017, Nogueira et al., 2016, Dai Nguyen et al., 2017]. Perhaps closer to our work is [Martinez-Cantin et al., 2018], considering function outliers; however, no specific corruption model was adopted, and no theoretical regret bounds were given.

In [Bogunovic et al., 2018a], bounds on the simple regret are given for the case that the final reported input is adversarially perturbed, whereas the selected inputs are only subject to random output noise. This makes it desirable to seek broad peaks, which bears some similarity to the input noise viewpoint [Beland and Nair, 2017, Nogueira et al., 2016, Dai Nguyen et al., 2017] and level-set estimation [Gotovos et al., 2013, Bogunovic et al., 2016b]. Our goal of attaining small cumulative regret under input perturbations requires very different techniques from these previous works. Another distinct notion of robustness is considered in [Bogunovic et al., 2018b], in which some experiments in a batch may fail to produce an outcome. None of the preceding works provide regret bounds in the case of non-stochastic corrupted observations.

Related work on corrupted bandits. Adversarially corrupted observations have recently been considered in the finite-arm stochastic MAB problem under various corruption models [Lykouris et al., 2018, Gupta et al., 2019, Kapoor et al., 2019]. As mentioned above, [Lykouris et al., 2018] adopted a "fast-slow" algorithmic approach; this led to regret bounds of the form $R_T = O(KC \cdot R_T^{\text{non-c}})$, where $R_T^{\text{non-c}}$ is a standard regret bound for the non-corrupted MAB setting. In [Gupta et al., 2019], this bound was improved to $O(KC + R_T^{\text{non-c}})$ using an epoch-based approach in which the estimates of the arms' means are reset after each epoch, and the previous epoch guides which arms are selected in the next one.

Our algorithmic approach is based on that of [Lykouris et al., 2018]; however, the bulk of the theoretical analysis requires novel ideas. In particular, our need to handle an infinite action space with correlated rewards between actions poses considerable challenges, as discussed above. In more detail, we note the following:

- Even when studying the case of a known corruption level (which is done as a stepping stone towards our main results), it is non-trivial to characterize the effect of the corruptions (see Lemma 2 below);
- Characterizing that certain suboptimal points are never sampled after a certain time requires significant technical effort (see Lemmas 7 and 8 below);
- We adopt a UCB-style approach (Alg. 2) complementary to the elimination-style approach of [Lykouris et al., 2018], and the former kind may be of independent interest even in the finite-arm setting.

In a parallel independent work [Li et al., 2019], cumulative regret bounds were given for stochastic linear bandits, which are a special case of the GP setting (with a linear kernel). The algorithm of [Li et al., 2019] is in fact more akin to that of [Gupta et al., 2019], which is potentially preferable due the latter attaining better bounds in the finite-arm setting. However, the algorithm and results of [Li et al., 2019] crucially rely on the notion of *gaps* between the function values of corner points in the domain, and the idea of exploiting these gaps for linear bandits has no apparent generalization to the GP setting with general kernels. In addition, even when we specialize to the linear kernel, neither our results nor those of [Li et al., 2019] imply each other, and the two both have benefits not provided by the other; see Appendix K for details.

Outline. We introduce the corruption-tolerant kernelized MAB problem in Section 2, and then present algorithms for three settings with increasing difficulty: Known corruption level (Section 3), simultaneous handing of no corruption and a known corruption level (Section 4), and unknown corruption level (Section 5).

2 Problem Statement

We consider the problem of sequentially maximizing a fixed unknown function $f : D \to [-B_0, B_0]$, where $D \subset \mathbb{R}^d$ is a compact set and $B_0 > 0$. We assume that D is endowed with a kernel function $k(\cdot,\cdot)$ defined on $D \times D$, and the kernel is normalized to satisfy $k(\boldsymbol{x}, \boldsymbol{x}') \leq 1$ for all $\boldsymbol{x}, \boldsymbol{x}' \in D$. We also assume that f has a bounded norm in the corresponding Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_k(D)$, i.e.,

$\|f\|_k \le B$. This assumption permits the construction of confidence bounds via Gaussian process (GP) methods (see Lemma 1 below).

In the non-corrupted setting, at every time step t, we choose $\boldsymbol{x}_t \in D$, and observe a noisy function value $y_t = f(\boldsymbol{x}_t) + \epsilon_t$. In this work, we consider the corrupted setting, where we only observe an adversarially corrupted sample $\tilde{y}_t$. Formally, for each $t = 1, \dots, T$:

- Based on the previous decisions and corresponding corrupted observations $\{(\boldsymbol{x}_i, \tilde{y}_i)\}_{i=1}^{t-1}$, the player selects a probability distribution $\Phi_t(\cdot)$ over D.
- Based on the knowledge of the true function f,[1] the previous decisions and corresponding observations $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{t-1}$, and the player's distribution $\Phi_t(\cdot)$, the adversary chooses the corruptions $c_t(\cdot) : D \to [-B_0, B_0]$.
- The agent draws $\boldsymbol{x}_t \in D$ at random from Φ_t, and observes the noisy and corrupted observation:

$$\tilde{y}_t = y_t + c_t(\boldsymbol{x}_t), \tag{1}$$

where y_t is the noisy non-corrupted observation: $y_t = f(\boldsymbol{x}_t) + \epsilon_t$, where $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$ with independence between times t.

Note that the adversary is allowed to be adaptive, i.e., the corruptions $c_t(\cdot)$ may depend on the agent's previously selected points and corresponding stochastic observations, as well as the distribution $\Phi_t(\cdot)$ of the player's next choice, but not its specific realization $\boldsymbol{x}_t$.

We say that the problem instance is *C-corrupted* (i.e., the *corruption level* is C) if

$$\sum_{t=1}^{T} \max_{\boldsymbol{x} \in D} |c_t(\boldsymbol{x})| \le C. \tag{2}$$

Clearly, when $C = 0$, we recover the standard non-corrupted setting. We measure the performance using the cumulative regret, which is also typically used in the non-corrupted bandit setting [Srinivas et al., 2010]:

$$R_T = \sum_{t=1}^{T} \big(f(\boldsymbol{x}^*) - f(\boldsymbol{x}_t)\big), \tag{3}$$

where $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in D} f(\boldsymbol{x})$. As noted in [Lykouris et al., 2018], one could alternatively define the cumulative regret with respect to the corrupted values $\{f(\boldsymbol{x}) + c_t(\boldsymbol{x})\}$; the two notions coincide to within at most $2C$, and such a difference will be negligible in our regret bound anyway. In Appendix C, we outline how our results can be adapted for simple regret (i.e., the regret of a point reported at the end of T rounds).

2.1 Standard (non-corrupted) setting

In the non-corrupted setting, existing algorithms use Gaussian likelihood models for the observations and zero-mean GP priors for modeling the uncertainty in f. Posterior updates are performed according to a "fictitious" model in which the noise variables $\epsilon_t = y_t - f(\boldsymbol{x}_t)$ are drawn independently across t from $\mathcal{N}(0, \lambda)$, where λ is a hyperparameter that may differ from the true noise variance σ^2. Given a sequence of inputs $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_t\}$ and their noisy observations $\{y_1, \dots, y_t\}$, the posterior distribution under this $\mathrm{GP}(\boldsymbol{0}, k)$ prior is also Gaussian, with the mean and variance

$$\mu_t(\boldsymbol{x}) = \boldsymbol{k}_t(\boldsymbol{x})^T \big(\boldsymbol{K}_t + \lambda \mathbf{I}_t\big)^{-1} \boldsymbol{y}_t, \tag{4}$$

$$\sigma_t^2(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}_t(\boldsymbol{x})^T \big(\boldsymbol{K}_t + \lambda \mathbf{I}_t\big)^{-1} \boldsymbol{k}_t(\boldsymbol{x}), \tag{5}$$

where $\boldsymbol{k}_t(\boldsymbol{x}) = \big[k(\boldsymbol{x}_i, \boldsymbol{x})\big]_{i=1}^{t}$, and $\boldsymbol{K}_t = \big[k(\boldsymbol{x}_t, \boldsymbol{x}_{t'})\big]_{t,t'}$ is the kernel matrix. Common kernels include the linear, squared exponential (SE) and Matérn kernels.

The main quantity that characterizes the regret bounds in the non-corrupted setting [Srinivas et al., 2010, Chowdhury and Gopalan, 2017] is the *maximum information gain*, defined at time t as

$$\gamma_t = \max_{\boldsymbol{x}_1, \dots, \boldsymbol{x}_t} \frac{1}{2} \ln \det(\boldsymbol{I}_t + \lambda^{-1} \boldsymbol{K}_t). \tag{6}$$

For compact and convex domains, γ_t is sublinear in t for various classes of kernels, e.g., $\mathcal{O}((\ln t)^{d+1})$ for the SE kernel, and $\mathcal{O}(t^{(d+1)d/((d+1)d+2\nu)} \ln t))$ for the Matérn kernel with $\nu > 1$ [Srinivas et al., 2010].

The following well-known result of [Abbasi-Yadkori, 2013] provides confidence bounds around the unknown function in the non-corrupted setting.

Lemma 1. *Fix $f \in \mathcal{H}_k(D)$ with $\|f\|_k \le B$, and consider the sampling model $y_t = f(\boldsymbol{x}_t) + \epsilon_t$, with independent noise $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$. Under the choice*

$$\beta_t = B + \sigma \lambda^{-1/2} \sqrt{2(\gamma_{t-1} + \ln(1/\delta))}, \tag{7}$$

the following holds with probability at least $1 - \delta$:

$$|\mu_{t-1}(\boldsymbol{x}) - f(\boldsymbol{x})| \le \beta_t \sigma_{t-1}(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in D, \forall t \ge 1, \tag{8}$$

where $\mu_{t-1}(\cdot)$ and $\sigma_{t-1}(\cdot)$ are given in (4) and (5).

This lemma follows directly from [Abbasi-Yadkori, 2013, Theorem 3.11] (and [Abbasi-Yadkori, 2013, Remark 3.13]) and the definition (6) of γ_t.

Lack of robustness against adversarial corruptions. In the noisy non-corrupted setting, several algorithms have been developed and analyzed. A particularly well-known example is GP-UCB, which selects $\boldsymbol{x}_t \in \arg\max_{\boldsymbol{x} \in D} \mathrm{ucb}_{t-1}(\boldsymbol{x}) := \mu_{t-1}(\boldsymbol{x}) + \beta_t \sigma_{t-1}(\boldsymbol{x})$. GP-UCB achieves sublinear cumulative regret with

[1] While knowing f may appear to make the adversary overly strong, the defense mechanism in [Lykouris et al., 2018] for the finite-arm setting also implicitly allows the adversary to know the reward distributions.

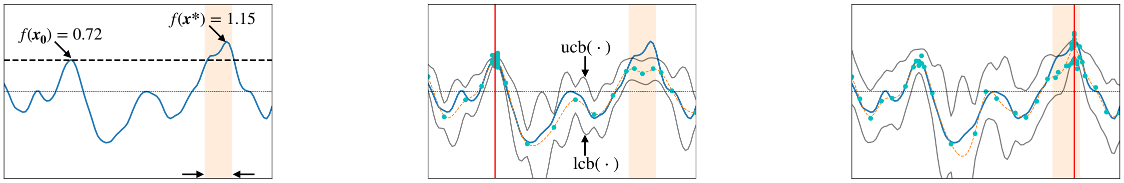

Figure 1: (Left) Function f, its global maximizer $\boldsymbol{x}^*$, a local maximizer $\boldsymbol{x}_0$, and the corruption region. (Middle) GP-UCB eliminates the optimal region (and $\boldsymbol{x}^*$) early on due to the corruptions, and continues sampling points in the suboptimal region around $\boldsymbol{x}_0$. (Right) Our corruption-aware algorithm (see Algorithm 1) does not eliminate the optimal region, and after the corruption budget is exhausted, it identifies the true maximizer $\boldsymbol{x}^*$.

high probability [Srinivas et al., 2010, Chowdhury and Gopalan, 2017], for a suitably chosen β_t (e.g., as in (7)). Despite this success in the non-corrupted setting, these algorithms can fail under adversarial corruptions.

An illustrative example is provided in Figure 1. Observations that correspond to the points sampled in the shaded region around the global maximizer $\boldsymbol{x}^*$ are corrupted by the value $-f(\boldsymbol{x}^*)/3$, up to a total corruption budget ($C = 3.5$). In Figure 1 (Middle), the points selected by GP-UCB for $t = 50$ time steps are shown. GP-UCB eliminates the global maximizer early on due to corruptions, and later on, it only selects points from the suboptimal region and consequently suffers linear cumulative regret. In the subsequent sections, we design algorithms that are robust to corruptions, and are able to identify the true maximizer after the corruption budget C is exhausted (see Figure 1 (Right)).

3 Known Corruption Setting

We first consider the case that the total corruption C in (2) is known. Given a sequence of inputs $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_t\}$ and their corrupted observations $\{\tilde{y}_1, \dots, \tilde{y}_t\}$ (with $\tilde{y}_i = y_i + c_i(\boldsymbol{x}_i)$), we form a posterior mean according to a $\mathrm{GP}(0, k(\boldsymbol{x}, \boldsymbol{x}'))$ prior and $\mathcal{N}(0, \lambda)$ sampling noise as follows:

$$\tilde{\mu}_t(\boldsymbol{x}) = \boldsymbol{k}_t(\boldsymbol{x})^T(\boldsymbol{K}_t + \lambda \boldsymbol{I})^{-1}\tilde{\boldsymbol{y}}_t, \tag{9}$$

where $\tilde{\boldsymbol{y}}_t = [\tilde{y}_1, \dots, \tilde{y}_t]$. Note that this matches the posterior mean formed in the non-corrupted setting, simply replacing $\boldsymbol{y}_t$ by $\tilde{\boldsymbol{y}}_t$. In addition, we form the same posterior standard deviation $\sigma_{t-1}(\boldsymbol{x})$ as in the non-corrupted setting. The role of the parameter λ is discussed in Appendix I.

The following lemma provides an upper bound on the difference between the non-corrupted and corrupted posterior means, and is proved using the definitions of μ_t and $\tilde{\mu}_t$ along with RKHS function properties. All proofs can be found in the supplementary material.

Lemma 2. *For any $\boldsymbol{x} \in D$ and $t \geq 1$, we have $|\mu_{t-1}(\boldsymbol{x}) - \tilde{\mu}_{t-1}(\boldsymbol{x})| \leq C\lambda^{-1/2}\sigma_{t-1}(\boldsymbol{x})$, where $\mu_{t-1}(\cdot)$ and $\sigma_{t-1}(\cdot)$ are given in (4) and (5), and $\tilde{\mu}_{t-1}(\cdot)$ is given in (9), with $\lambda > 0$.*

Algorithm 1 Gaussian Process UCB algorithm with known total corruption C

```
Input: Prior GP(0, k), parameters $\sigma$, $\lambda$, $B$, $\{\beta_t\}_{t \geq 1}$,
    and total corruption $C$
for $t = 1, 2, \dots, T$ do
    Set
        $\boldsymbol{x}_t = \arg\max_{\boldsymbol{x} \in D} \tilde{\mu}_{t-1}(\boldsymbol{x}) + \beta_t^{(A_1)}\sigma_{t-1}(\boldsymbol{x}),$    (13)
     where $\beta_t^{(A_1)} = \beta_t + \lambda^{-1/2}C$
    Observe $\tilde{y}_t$ obtained via $\tilde{y}_t = f(\boldsymbol{x}_t) + \epsilon_t + c_t(\boldsymbol{x}_t)$
    Update $\tilde{\mu}_t$ and $\sigma_t$ according to (9) and (5) by
    including $(\boldsymbol{x}_t, \tilde{y}_t)$
end for
```

By combining Lemmas 1 and 2, we obtain the following.

Lemma 3. *Fix $f \in \mathcal{H}_k(D)$ with $\|f\|_k \leq B$. Under the choice $\beta_t^{(A_1)} = \beta_t + \lambda^{-1/2}C$ with β_t given in (7) and $\lambda > 0$, we have with probability at least $1 - \delta$ that*

$$|\tilde{\mu}_{t-1}(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \beta_t^{(A_1)}\sigma_{t-1}(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in D, \forall t \geq 1, \tag{10}$$

where $\tilde{\mu}_{t-1}$ and σ_{t-1} are given in (9) and (5).

In Algorithm 1 (A_1), we present an upper confidence bound based algorithm with enlarged confidence bounds in accordance with Lemma 3. We explicitly define these confidence bounds as follows:

$$\mathrm{ucb}_t^{(A_1)}(\boldsymbol{x}) = \tilde{\mu}_t(\boldsymbol{x}) + \beta_{t+1}^{(A_1)}\sigma_t(\boldsymbol{x}), \tag{11}$$

$$\mathrm{lcb}_t^{(A_1)}(\boldsymbol{x}) = \tilde{\mu}_t(\boldsymbol{x}) - \beta_{t+1}^{(A_1)}\sigma_t(\boldsymbol{x}). \tag{12}$$

Once the validity of these confidence bounds is established via (10), one can use standard analysis techniques [Srinivas et al., 2010] to bound the cumulative regret. This is formally stated in the following.

Lemma 4. *Under the choice of $\beta_t^{(A_1)}$ in Lemma 3 and $\lambda = 1$, conditioned on the event (10), the cumulative regret incurred by Algorithm 1 satisfies $R_T = \mathcal{O}\big((B + C + \sqrt{\ln(1/\delta)})\sqrt{\gamma_T T} + \gamma_T\sqrt{T}\big)$.*

The main theorem of this section is now obtained via a direct combination of Lemmas 3 and 4.

Theorem 5. *In the C-corrupted setting, Algorithm 1 with $\lambda = 1$ and $\beta_t^{(A_1)}$ set as in Lemma 3, attains, with probability at least $1-\delta$, cumulative regret $R_T = \mathcal{O}\big(\big(B + C + \sqrt{\ln(1/\delta)}\big)\sqrt{\gamma_T T} + \gamma_T\sqrt{T}\big)$.*

Note that when $C = 0$, this result recovers known non-corrupted cumulative regret bounds (*cf.* [Chowdhury and Gopalan, 2017, Theorem 3]). More generally, we can decompose the obtained regret bound into two terms: R_T behaves as

$$\mathcal{O}\Big(\underbrace{C\sqrt{\gamma_T T}}_{\text{due to corruption}} + \underbrace{\big(B + \sqrt{\ln(1/\delta)}\big)\sqrt{\gamma_T T} + \gamma_T\sqrt{T}}_{\text{non-corrupted regret bound}} \Big). \tag{14}$$

The obtained regret bound can be made more explicit by substituting the bound on γ_T for particular kernels [Srinivas et al., 2010], e.g., for the SE kernel we obtain $R_T = \mathcal{O}\big((C+B)\sqrt{T(\log T)^{d+1}} + (\log T)^{d+1}\sqrt{T}\big)$. In Appendix J, we argue that the linear dependence on C is unavoidable for any algorithm, and discuss cases where the dependence on T is near-optimal. However, we do not necessarily claim that the *joint* dependence on C and T is optimal; this is left for future work.

4 Known-or-Zero Corruption Setting

In the previous section, we assumed that the upper bound C on the total corruption is known and the problem is C-corrupted. In this section, we also assume that C is known, but we consider a scenario in which the problem may be either C-corrupted or non-corrupted (i.e., the standard setting). Our goal is to develop an algorithm that has a similar guarantee to the previous section in the corrupted case, while also attaining a similar guarantee to GP-UCB [Srinivas et al., 2010] in the non-corrupted case, and thus obtaining strong guarantees in the two settings *simultaneously.* Theorem 5 fails to achieve this goal, since the regret depends on C even if the problem is non-corrupted.

Our algorithm Fast-Slow GP-UCB is described in Algorithm 2. It makes use of two instances labeled F (fast; Line 6) and S (slow; Line 8). The S instance is played with probability $1/C$, while the rest of the time F is played. The intuition is that F shrinks the confidence bounds faster but is not robust to corruptions, while S is slower but robust to corruptions. We formalize this intuition below in Lemma 6 and (20)–(21).

Algorithm 2 Fast-Slow GP-UCB algorithm

```
Input: Prior GP$(0,k)$, parameters $\sigma$, $\lambda$, $B$, $\alpha$, $\{\beta_t^{(F)}\}_{t\ge1}$, $\{\beta_t^{(S)}\}_{t\ge1}$, and total corruption $C$
1: Initialize: $t_S, t_F := 1$, isValid = True
2: for $t = 1, 2, \dots, T$ do
3:   if isValid is True then
4:     Sample instance $A_t$: $A_t = S$ with probability $\min\{1, C^{-1}\}$. Otherwise, $A_t = F$.
5:     if $A_t = F$ then
6:       $\boldsymbol{x}_t \leftarrow \arg\max_{\boldsymbol{x}\in D} \min_{A\in\{F,S\}} \overline{\mathrm{ucb}}^{(A)}_{t_A-1}(\boldsymbol{x};1)$
7:     else
8:       $\boldsymbol{x}_t \leftarrow \arg\max_{\boldsymbol{x}\in D} \mathrm{ucb}^{(S)}_{t_S-1}(\boldsymbol{x};\alpha)$
9:     Observe: $\tilde{y}_t = f(\boldsymbol{x}_t) + c_t(\boldsymbol{x}_t) + \epsilon_t$
10:    Set: $t_{A_t} \leftarrow t_{A_t} + 1$
11:    Update: $\tilde{\mu}^{(A_t)}(\cdot)$, $\sigma^{(A_t)}(\cdot)$ to time $t_{A_t}$ by including $(\boldsymbol{x}_t, \tilde{y}_t)$
12:    if $\min_{\boldsymbol{x}} \big\{\overline{\mathrm{ucb}}^{(F)}_{t_F-1}(\boldsymbol{x};1) - \overline{\mathrm{lcb}}^{(S)}_{t_S-1}(\boldsymbol{x};1)\big\} < 0$ then
13:      isValid $\leftarrow$ False
14:  else
15:    Use all the collected data $\{\boldsymbol{x}_i, \tilde{y}_i\}_{i=1}^t$ to compute $\tilde{\mu}_{t-1}(\cdot)$ and $\sigma_{t-1}(\cdot)$
16:    Choose next point, observe and update according to Algorithm 1
```

The instances use the following confidence bounds depending on an exploration parameter $\beta^{(A)}_{t_A+1}$ and an additional parameter $\alpha > 1$ whose role is discussed in Appendix I and after Lemma 8 below:

$$\mathrm{ucb}^{(A)}_{t_A}(\boldsymbol{x};\alpha) = \tilde{\mu}^{(A)}_{t_A}(\boldsymbol{x}) + \alpha\beta^{(A)}_{t_A+1}\sigma^{(A)}_{t_A}(\boldsymbol{x}) \tag{15}$$

$$\mathrm{lcb}^{(A)}_{t_A}(\boldsymbol{x};\alpha) = \tilde{\mu}^{(A)}_{t_A}(\boldsymbol{x}) - \alpha\beta^{(A)}_{t_A+1}\sigma^{(A)}_{t_A}(\boldsymbol{x}), \tag{16}$$

where t_A is the number of times an instance $A \in \{F, S\}$ has been selected at a given time instant. We also make use of the following *intersected* confidence bounds, which have the convenient feature of being monotone:

$$\overline{\mathrm{ucb}}^{(A)}_{t_A-1}(\boldsymbol{x};\alpha) = \min_{t'_A \le t_A} \mathrm{ucb}^{(A)}_{t'_A-1}(\boldsymbol{x};\alpha), \tag{17}$$

$$\overline{\mathrm{lcb}}^{(A)}_{t_A-1}(\boldsymbol{x};\alpha) = \max_{t'_A \le t_A} \mathrm{lcb}^{(A)}_{t'_A-1}(\boldsymbol{x};\alpha). \tag{18}$$

In Fast-Slow GP-UCB, we check if the following condition (Line 12) holds:

$$\min_{\boldsymbol{x}\in D} \Big\{ \overline{\mathrm{ucb}}^{(F)}_{t_F-1}(\boldsymbol{x};1) - \overline{\mathrm{lcb}}^{(S)}_{t_S-1}(\boldsymbol{x};1) \Big\} < 0. \tag{19}$$

In the non-corrupted setting, under the high-probability event in Lemma 1 (for both F and S), this condition never holds. Hence, when it does hold, we have detected that the problem is C-corrupted.In such a case, the algorithm permanently switches to running Algorithm 1 with C as the input. Note that we can check the condition in (19) by using a global optimizer to find a minimizer of $g(\boldsymbol{x}) := \overline{\mathrm{ucb}}^{(F)}_{t_F-1}(\boldsymbol{x};1) - \overline{\mathrm{lcb}}^{(S)}_{t_S-1}(\boldsymbol{x};1)$, and checking whether its value is smaller than 0.

Finally, the inner minimization over $A \in \{F, S\}$ in the F instance, together with the validity of the condition (19), ensures that F does not select a point that is already "ruled out" by the robust instance S. We make this statement precise in Lemma 7 below.

4.1 Analysis

First, we provide a high-probability bound on the total corruption that is observed by the S instance. Specifically, we show that because it is sampled with probability $1/C$, the total corruption observed by S is constant with high probability, i.e., it is upper bounded by a value not depending on T.

Lemma 6. *The S instance in* Fast-Slow GP-UCB *observes, with probability at least $1-\delta$, a total corruption $\sum_{t=1}^{T}|c_t(\boldsymbol{x}_t)|\mathbb{1}\{A_t = S\}$ of at most $3 + B_0\ln(1/\delta)$.*

We now fix a constant $\delta \in (0,1)$ and condition on three high-probability events:

1. If $\beta^{(F)}_{t_F} = B+\sigma\lambda^{-1/2}\sqrt{2\left(\gamma_{t_F-1}+\ln\left(\frac{5}{\delta}\right)\right)}$ and the setting is non-corrupted, the following holds with probability at least $1-\frac{\delta}{5}$:
$$\mathrm{lcb}^{(F)}_{t_F-1}(\boldsymbol{x};1) \le f(\boldsymbol{x}) \le \mathrm{ucb}^{(F)}_{t_F-1}(\boldsymbol{x};1), \qquad (20)$$
for all $\boldsymbol{x} \in D$ and $t_F \ge 1$. This claim follows from Lemma 1 by setting the corresponding failure probability to $\frac{\delta}{5}$.

2. If $\beta^{(S)}_{t_S} = B + \sigma\lambda^{-1/2}\sqrt{2\left(\gamma_{t_S-1}+\ln\left(\frac{5}{\delta}\right)\right)} + \lambda^{-1/2}(3+B_0\ln\left(\frac{5}{\delta}\right))$, then the following holds in both the non-corrupted and corrupted settings with probability at least $1-\frac{2\delta}{5}$:
$$\mathrm{lcb}^{(S)}_{t_S-1}(\boldsymbol{x};1) \le f(\boldsymbol{x}) \le \mathrm{ucb}^{(S)}_{t_S-1}(\boldsymbol{x};1), \qquad (21)$$
for all $\boldsymbol{x} \in D$ and $t_S \ge 1$. This follows from Lemmas 3 and 6 (using $3 + B_0\ln\left(\frac{5}{\delta}\right)$ in place of C in Lemma 3), by setting the corresponding failure probabilities to $\frac{\delta}{5}$ in both. Taking the union bound over these two events establishes the claim. Note that (21) corresponds to $\alpha = 1$, but directly implies an analogous condition for all $\alpha > 1$ (since increasing α widens the confidence region (15)–(16)).

3. If the condition in (19) is detected at any time instant, then Algorithm 2 permanently switches to running Algorithm 1. If Algorithm 1 is run with $\beta^{(A_1)}_t = B+\sigma\lambda^{-1/2}\sqrt{2\left(\gamma_{t-1}+\ln\left(\frac{5}{\delta}\right)\right)}+\lambda^{-1/2}C$, then with probability at least $1-\frac{\delta}{5}$:
$$\mathrm{lcb}^{(A_1)}_{t-1}(\boldsymbol{x}) \le f(\boldsymbol{x}) \le \mathrm{ucb}^{(A_1)}_{t-1}(\boldsymbol{x}), \qquad (22)$$
for all $\boldsymbol{x} \in D$ and $t \ge 1$, under the definitions in (11). This is by Lemma 3 with $\frac{\delta}{5}$ in place of δ.

By the union bound, (20)–(22) all hold with probability at least $1-\frac{4\delta}{5}$. In addition, by the definitions in (17), these properties remain true when $\mathrm{ucb}^{(A)}$ and $\mathrm{lcb}^{(A)}$ are replaced by $\overline{\mathrm{ucb}}^{(A)}$ and $\overline{\mathrm{lcb}}^{(A)}$.

The confidence bounds of F are only valid in the non-corrupted case, and hence, in the case of corruptions we rely on the confidence bounds of S. Specifically, we show that F never queries a point that is strictly suboptimal according to the confidence bounds of S.

Lemma 7. *Suppose that* (20) *and* (21) *hold. For any time $t \ge 1$, if $A_t = F$ in* Fast-Slow GP-UCB*, then the selected point $\boldsymbol{x}_t \notin \mathcal{S}_{t_S}$, where*
$$\mathcal{S}_{t_S} = \{\boldsymbol{x} \in D :\ \exists \boldsymbol{x}' \in D,\ \overline{\mathrm{lcb}}^{(S)}_{t_S-1}(\boldsymbol{x}';1) > \overline{\mathrm{ucb}}^{(S)}_{t_S-1}(\boldsymbol{x};1)\} \qquad (23)$$
represents the set of strictly suboptimal points according to the intersected S-confidence bounds.

By the monotonicity of $\overline{\mathrm{lcb}}^{(S)}_{t_S-1}$ and $\overline{\mathrm{ucb}}^{(S)}_{t_S-1}$, the set $\mathcal{S}_{t_S}$ is non-shrinking in t. The proof shows that F always favors $\boldsymbol{x}'$ from (23) over $\boldsymbol{x} \in \mathcal{S}_{t_S}$, i.e., $\boldsymbol{x}'$ has a higher value of $\min_{A\in\{F,S\}}\overline{\mathrm{ucb}}^{(A)}_{t_A-1}(\cdot;1)$ (see Line 6 of Algorithm 2). To show this, we upper bound $\min_{A\in\{F,S\}}\overline{\mathrm{ucb}}^{(A)}_{t_A-1}(\boldsymbol{x};1)$ in terms of $\overline{\mathrm{lcb}}^{(S)}_{t_S-1}$ via (23), and lower bound $\min_{A\in\{F,S\}}\overline{\mathrm{ucb}}^{(A)}_{t_A-1}(\boldsymbol{x};1)$ in terms of $\overline{\mathrm{lcb}}^{(S)}_{t_S-1}$ via the confidence bounds and condition (19).

The next lemma characterizes the number of queries made by the S instance before a suboptimal point becomes "eliminated", i.e., the time after which the point belongs to $\mathcal{S}_{t_S}$.

Lemma 8. *Suppose that the S instance is run with $\beta^{(S)}_{t_S}$ corresponding to* (21) *and $\alpha = 2$. Then, conditioned on the high-probability confidence bounds in* (21)*, for any given suboptimal point $\boldsymbol{x} \in D$ such that $f(\boldsymbol{x}^*) - f(\boldsymbol{x}) \ge \Delta_0 > 0$, it holds that $\boldsymbol{x} \in \mathcal{S}_{t_S}$ after*
$$t_S = \min\left\{\tau \,:\, \sqrt{\tfrac{16\alpha^2(\beta^{(S)}_\tau)^2\gamma_\tau}{\tau}} \le \tfrac{\Delta_0}{10}\right\}. \qquad (24)$$

This lemma's proof is perhaps the trickiest, and crucially relies on the fact that $\alpha > 1$. We show that by the time given in (24), we have encountered a round i in which a $\frac{\Delta_0}{10}$-optimal point $\boldsymbol{x}_i$ is queried with the confidence width also being at most $\frac{\Delta_0}{10}$. This means that $\boldsymbol{x}_i$ is much closer to optimal than the Δ_0-suboptimal point $\boldsymbol{x}$ in the lemma statement. Using the fact that $\boldsymbol{x}_i$ had a higher UCB score than $\boldsymbol{x}$, we can also deduce that the posterior standard deviation at $\boldsymbol{x}$ was not too large. Since replacing $\alpha = 2$ by $\alpha = 1$ (as done in the definition of $\mathcal{S}_{t_S}$ in (23)) halves the confidence width, we can combine the above findings to deduce that the confidence bounds indeed rule out $\boldsymbol{x}$ at time $i < t_S$, and hence also for all subsequent times due to the monotonicity of the confidence bounds.

Finally, we state the main theorem of this section, whose proof combines the preceding lemmas.

Theorem 9. *For any $f \in \mathcal{H}_k(D)$ with $\|f\|_k \leq B$, let $\delta \in (0,1)$, and consider* Fast-Slow GP-UCB *run with $\alpha = 2$, $\lambda = 1$,*

$$\beta^{(F)}_{t_F} = B + \sigma\sqrt{2\left(\gamma_{t_F-1} + \ln\left(\tfrac{5}{\delta}\right)\right)}, \tag{25}$$

$$\beta^{(S)}_{t_S} = B + \sigma\sqrt{2\left(\gamma_{t_S-1} + \ln\left(\tfrac{5}{\delta}\right)\right)} + \left(3 + B_0 \ln\left(\tfrac{5}{\delta}\right)\right), \tag{26}$$

and $\beta^{(A_1)}_t$ set in Algorithm 1 as $\beta^{(A_1)}_t = B + \sigma\sqrt{2\left(\gamma_{t-1} + \ln\left(\frac{5}{\delta}\right)\right)} + C$. Then, after T rounds, with probability at least $1-\delta$ the cumulative regret satisfies

$$R_T = \mathcal{O}\left(\left(B + B_0 \ln(\tfrac{1}{\delta}) + \sqrt{\ln(\tfrac{1}{\delta})}\right)\sqrt{T\gamma_T} + \gamma_T\sqrt{T}\right) \tag{27}$$

in the non-corrupted case and

$$R_T = \mathcal{O}\bigg((1+C)\ln(\tfrac{T}{\delta})\Big(\Big(B + B_0\ln(\tfrac{1}{\delta}) + \sqrt{\ln(\tfrac{1}{\delta})}\Big) \times \sqrt{\gamma_T T} + \gamma_T\sqrt{T}\Big)\bigg) \tag{28}$$

in the corrupted case.

The non-corrupted case is straightforward to prove, essentially applying standard arguments separately to F and S. The corrupted case requires more effort. Lemma 8 characterizes the time after which points with a given regret are no longer sampled by S, which permits bounding the cumulative regret incurred by S. By Lemma 7, the points in $\mathcal{S}_{t_S}$ are also not sampled by F, and on average F is played at most C times more frequently than S. Converting this average to a high-probability bound using basic concentration, this factor of C becomes $C\ln(\frac{5T}{\delta})$, and we obtain (28).

Using the notation $\tilde{\mathcal{O}}(\cdot)$ to hide logarithmic factors, the bound obtained in the non-corrupted case simplifies to $R_T = \tilde{\mathcal{O}}\big((B+B_0)\sqrt{T\gamma_T} + \gamma_T\sqrt{T}\big)$, and unlike the result from Theorem 5, it does not depend on C. The obtained bound is only a constant factor away from the standard non-corrupted one (*cf.* (14)), while at the same time our algorithm achieves $R_T = \tilde{\mathcal{O}}\big(C(B + B_0)\sqrt{T\gamma_T} + C\gamma_T\sqrt{T}\big)$ in the C-corrupted case. As before, we can make the results obtained in this theorem more explicit by substituting the bounds for γ_T for various kernels of interest [Srinivas et al., 2010].

5 Unknown Corruption Setting

In this section, we assume that the total corruption C defined in (2) is unknown to the algorithm. Despite this additional challenge, most of the details are similar to the known-or-zero setting, so to avoid repetition, we omit some details and focus on the key differences.

Algorithm. Our corruption-agnostic algorithm is shown in Algorithm 3. We again take inspiration from the finite-arm counterpart [Lykouris et al., 2018], considering *layers* $\ell = 1, \dots, \lceil \log_2 T \rceil$ that are sampled with probability $2^{-\ell}$ (with any remaining probability going to layer 1). The idea is that any layer with $2^\ell \geq C$ is robust, for the same reason that the S instance is robust in Fast-Slow GP-UCB (Algorithm 2).

Each instance ℓ makes use of confidence bounds defined as follows for some parameters $\beta^{(\ell)}_{t_\ell}$ to be chosen later:

$$\mathrm{ucb}^{(\ell)}_{t_\ell}(\boldsymbol{x};\alpha) = \tilde{\mu}_{t_\ell}(\boldsymbol{x}) + \alpha\beta^{(\ell)}_{t_\ell+1}\sigma_{t_\ell}(\boldsymbol{x}) \tag{29}$$

$$\mathrm{lcb}^{(\ell)}_{t_\ell}(\boldsymbol{x};\alpha) = \tilde{\mu}_{t_\ell}(\boldsymbol{x}) - \alpha\beta^{(\ell)}_{t_\ell+1}\sigma_{t_\ell}(\boldsymbol{x}), \tag{30}$$

where t_ℓ denotes the number of times instance ℓ has been selected by time t, and $\alpha > 1$. Similarly to the Section 4, we define intersected confidence bounds:

$$\overline{\mathrm{ucb}}^{(\ell)}_{t_\ell-1}(\boldsymbol{x};\alpha) = \min_{t'_\ell \leq t_\ell} \mathrm{ucb}^{(\ell)}_{t'_\ell-1}(\boldsymbol{x};\alpha) \tag{31}$$

$$\overline{\mathrm{lcb}}^{(\ell)}_{t_\ell-1}(\boldsymbol{x};\alpha) = \max_{t'_\ell \leq t_\ell} \mathrm{lcb}^{(\ell)}_{t'_\ell-1}(\boldsymbol{x};\alpha). \tag{32}$$

Each instance ℓ selects a point according to $\arg\max_{\boldsymbol{x}\in M^{(\ell)}_t} \mathrm{ucb}^{(\ell)}_{t_\ell-1}(\boldsymbol{x};\alpha)$, where $M^{(\ell)}_t$ represents a set of *potential maximizers* at time t, i.e., a set of points that could still be the global maximizer according to the confidence bounds. More formally, these sets are defined recursively as follows:[2]

$$M^{(\ell)}_t := \Big\{\boldsymbol{x} \in D : \overline{\mathrm{ucb}}^{(\ell)}_{t_\ell-1}(\boldsymbol{x};1) \geq \max_{\boldsymbol{x}'\in D} \overline{\mathrm{lcb}}^{(\ell)}_{t_\ell-1}(\boldsymbol{x}';1)\Big\} \quad \text{for } \ell = \lceil \log_2 T \rceil, \tag{33}$$

$$M^{(\ell)}_t := M^{(\ell+1)}_t \cap \Big\{\boldsymbol{x} \in D : \overline{\mathrm{ucb}}^{(\ell)}_{t_\ell-1}(\boldsymbol{x};1) \geq \max_{\boldsymbol{x}'\in D} \overline{\mathrm{lcb}}^{(\ell)}_{t_\ell-1}(\boldsymbol{x}';1)\Big\} \quad \text{for } \ell < \lceil \log_2 T \rceil. \tag{34}$$

Two key properties of these sets are: (i) $M^{(\ell)}_t \subseteq M^{(\ell)}_{t'}$ for every $t > t'$ and $\ell \in \{1, \dots, \lceil \log_2 T \rceil\}$ due to the monotonicity of the confidence bounds; and (ii) $M^{(1)}_t \subseteq M^{(2)}_t \cdots \subseteq M^{(\lceil \log_2 T \rceil)}_t$ for every t. The latter property

[2] Note that a given set $M^{(\ell)}_t$ may be non-convex, making the constraint $\boldsymbol{x} \in D$ in the UCB rule non-trivial to enforce in practice (e.g., one may use a discretization argument). Our focus is on the theory, in which we assume that the acquisition function can be optimized exactly.

Algorithm 3 FAST-SLOW GP-UCB algorithm with Unknown Corruption Level C

```
Input: Prior $\mathrm{GP}(0,k)$, parameters $\sigma$, $\lambda$, $B$, $\alpha$, $\{\beta^{(\ell)}_{t_\ell}\}_{t\geq 1}$ for all $\ell \in \{1,\dots,\lceil \log_2 T\rceil\}$
  Initialize: For all $\ell \in \{1,\dots,\lceil \log_2 T\rceil\}$, set $M^{(\ell)}_1 = D$
  for $t = 1, 2, \dots, T$ do
    Sample instance $\ell \in \{1,\dots,\lceil \log_2 T\rceil\}$ w.p. $2^{-\ell}$. With remaining prob., sample $\ell = 1$.
    if $M^{(\ell)}_t \neq \emptyset$ then
      $\boldsymbol{x}_t \leftarrow \arg\max_{\boldsymbol{x}\in M^{(\ell)}_t} \mathrm{ucb}^{(\ell)}_{t_\ell-1}(\boldsymbol{x};\alpha)$
      Observe $\tilde{y}_t = f(\boldsymbol{x}_t) + c_t(\boldsymbol{x}_t) + \epsilon_t$
      Update $\tilde{\mu}^{(\ell)}(\cdot)$, $\sigma^{(\ell)}(\cdot)$ by including $(\boldsymbol{x}_t, \tilde{y}_t)$
      $t_\ell \leftarrow t_\ell + 1$
      $M^{(\ell)}_{t+1} \leftarrow \{\boldsymbol{x} \in D : \overline{\mathrm{ucb}}^{(\ell)}_{t_\ell-1}(\boldsymbol{x};1) \geq \max_{\boldsymbol{x}'\in D} \overline{\mathrm{lcb}}^{(\ell)}_{t_\ell-1}(\boldsymbol{x}';1)\}$
      $M^{(i)}_{t+1} \leftarrow M^{(\ell)}_{t+1} \cap M^{(i)}_t$ for $i \in \{1,\dots,\ell-1\}$
      $M^{(i)}_{t+1} \leftarrow M^{(i)}_t$ for $i \in \{\ell+1,\dots,\lceil \log T\rceil\}$
    else
      $\ell \leftarrow \arg\min_{i\in\{\ell+1,\dots,\lceil \log_2 T\rceil\}}\{M^{(i)}_t \neq \emptyset\}$
      $\boldsymbol{x}_t \leftarrow \arg\max_{\boldsymbol{x}\in M^{(\ell)}_t} \mathrm{ucb}^{(\ell)}_{t_\ell-1}(\boldsymbol{x};\alpha)$
      Observe: $\tilde{y}_t = f(\boldsymbol{x}_t) + c_t(\boldsymbol{x}_t) + \epsilon_t$
      $M^{(i)}_{t+1} \leftarrow M^{(i)}_t$ for every $i \in \{1,\dots,\lceil \log_2 T\rceil\}$
```

implies that once a point is eliminated at some layer ℓ, it is also eliminated from all $M^{(1)}_t,\dots,M^{(\ell-1)}_t$, while the former property ensures that it remains eliminated for all subsequent time steps $\{t+1,\dots,T\}$.

Similarly to FAST-SLOW GP-UCB, each layer uses $\mathrm{ucb}^{(\ell)}_{t_\ell-1}(\boldsymbol{x};\alpha)$ *with α strictly larger than* 1 (e.g., $\alpha = 2$ suffices) in its acquisition function, while replacing α by 1 in the confidence bounds when constructing the set of potential maximizers. This is done to permit the application of Lemma 8; the intuition behind doing so is discussed in Appendix I and following Lemma 8.

In the case that $M^{(\ell)}_t$ corresponding to the selected ℓ at time t is empty, the algorithm finds the lowest layer i for which $M^{(i)} \neq \emptyset$, and selects the point that maximizes that layer's upper confidence bound. In this case, the algorithm makes no changes to the confidence bounds or the sets of potential maximizers.

Regret bound. With FAST-SLOW GP-UCB and its theoretical analysis in place, we can also obtain a near-identical regret bound in the case of unknown C. We only provide a brief outline here, with further details in the supplementary material.

We let the robust layer $\ell^* = \lceil \log_2 C\rceil$ play the role of F and eliminate suboptimal points. Since $2^{-\ell^*} \geq \frac{1}{2C}$, the regret incurred in the lower layers is at most a factor $2C$ higher than that of layer ℓ^* on average, and this leads to a similar analysis to that used in the proof of Theorem 9. Our final main result is stated as follows.

Theorem 10. *For any $f \in \mathcal{H}_k(D)$ with $\|f\|_k \leq B$, and any $\delta \in (0,1)$, under the parameters*

$$\beta^{(\ell)}_{t_\ell} = B + \sigma\sqrt{2\left(\gamma_{t_\ell-1} + \ln\left(\frac{4(1+\log_2 T)}{\delta}\right)\right)} + 3 + B_0 \ln\left(\frac{4(1+\log_2 T)}{\delta}\right), \quad (35)$$

we have that for any unknown corruption level $C > 0$, the cumulative regret of Algorithm 3 satisfies

$$R_T = \mathcal{O}\bigg((1+C)\ln(\tfrac{T}{\delta}) \times \bigg(\Big(B + B_0\ln(\tfrac{\log T}{\delta}) + \sqrt{\ln(\tfrac{\log T}{\delta})}\Big)\sqrt{\gamma_T T} + \gamma_T\sqrt{T}\bigg)\bigg) \quad (36)$$

with probability at least $1-\delta$.

This has the same form as (28), with $\frac{\delta}{\log T}$ in place of δ (since there are $\lceil \log_2 T\rceil$ layers).

6 Conclusion

We have introduced the kernelized MAB problem with adversarially corrupted samples. We provided novel algorithms based on enlarged confidence bounds and randomly-selected fast/slow instances that are provably robust against such corruptions, with the regret bounds being linear in the corruption level. To our knowledge, we are the first to handle this form of adversarial corruption in any bandit problem with an infinite action space and correlated rewards, which are two key notions that significantly complicate the analysis.

An immediate direction for further research is to better understand the *joint* dependence on the corruption level C and time horizon T. The linear $O(C)$ dependence is unavoidable (see Appendix J), and the $O(B\sqrt{\gamma_T T} + \gamma_T\sqrt{T})$ dependence matches well-known bounds for the non-corrupted setting [Srinivas et al., 2010, Chowdhury and Gopalan, 2017] (in some cases having near-matching lower bounds [Scarlett et al., 2017]), but it is unclear whether the *product* of these two terms is unavoidable.

Acknowledgments

This work was gratefully supported by Swiss National Science Foundation, under the grant SNSF NRP 75 407540_167189, ERC grant 815943 and ETH Zürich Postdoctoral Fellowship 19-2 FEL-47. J. Scarlett was supported by the Singapore National Research Foundation (NRF) under grant number R-252-000-A74-281.

References

[Abbasi-Yadkori, 2013] Abbasi-Yadkori, Y. (2013). Online learning for linearly parametrized control problems.

[Abbasi-Yadkori et al., 2011] Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320.

[Beland and Nair, 2017] Beland, J. J. and Nair, P. B. (2017). Bayesian optimization under uncertainty. NIPS BayesOpt 2017 workshop.

[Beygelzimer et al., 2011] Beygelzimer, A., Langford, J., Li, L., Reyzin, L., and Schapire, R. (2011). Contextual bandit algorithms with supervised learning guarantees. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 19–26.

[Bogunovic et al., 2016a] Bogunovic, I., Scarlett, J., and Cevher, V. (2016a). Time-varying Gaussian process bandit optimization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 314–323.

[Bogunovic et al., 2018a] Bogunovic, I., Scarlett, J., Jegelka, S., and Cevher, V. (2018a). Adversarially robust optimization with Gaussian processes. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5760–5770.

[Bogunovic et al., 2016b] Bogunovic, I., Scarlett, J., Krause, A., and Cevher, V. (2016b). Truncated variance reduction: A unified approach to Bayesian optimization and level-set estimation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1507–1515.

[Bogunovic et al., 2018b] Bogunovic, I., Zhao, J., and Cevher, V. (2018b). Robust maximization of non-submodular objectives. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 890–899.

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

[Dai Nguyen et al., 2017] Dai Nguyen, T., Gupta, S., Rana, S., and Venkatesh, S. (2017). Stable Bayesian optimization. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 578–591. Springer.

[Djolonga et al., 2013] Djolonga, J., Krause, A., and Cevher, V. (2013). High-dimensional Gaussian process bandits. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1025–1033.

[Freedman et al., 1975] Freedman, D. A. et al. (1975). On tail probabilities for martingales. *Annals of Probability*, 3(1):100–118.

[Gotovos et al., 2013] Gotovos, A., Casati, N., Hitz, G., and Krause, A. (2013). Active learning for level set estimation. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1344–1350.

[Gupta et al., 2019] Gupta, A., Koren, T., and Talwar, K. (2019). Better algorithms for stochastic bandits with adversarial corruptions. In *Conference on Learning Theory (COLT)*.

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

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

[Kandasamy et al., 2017] Kandasamy, K., Dasarathy, G., Schneider, J., and Póczos, B. (2017). Multi-fidelity bayesian optimisation with continuous approximations. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1799–1808. JMLR. org.

[Kandasamy et al., 2015] Kandasamy, K., Schneider, J., and Póczos, B. (2015). High dimensional Bayesian optimisation and bandits via additive models. In *International Conference on Machine Learning (ICML)*, pages 295–304.

[Kapoor et al., 2019] Kapoor, S., Patel, K. K., and Kar, P. (2019). Corruption-tolerant bandit learning. *Machine Learning*, 108(4):687–715.

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

[Li et al., 2019] Li, Y., Lou, E. Y., and Shan, L. (2019). Stochastic linear optimization with adversarial corruption. *arXiv preprint arXiv:1909.02109*.

[Lizotte et al., 2007] Lizotte, D. J., Wang, T., Bowling, M. H., and Schuurmans, D. (2007). Automatic gait optimization with Gaussian process regression. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 944–949.

[Lykouris et al., 2018] Lykouris, T., Mirrokni, V., and Paes Leme, R. (2018). Stochastic bandits robust to adversarial corruptions. In *ACM Symposium on Theory of Computing (STOC)*, pages 114–122. ACM.

[Martinez-Cantin et al., 2018] Martinez-Cantin, R., Tee, K., and McCourt, M. (2018). Practical Bayesian optimization in the presence of outliers. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

[Nogueira et al., 2016] Nogueira, J., Martinez-Cantin, R., Bernardino, A., and Jamone, L. (2016). Unscented Bayesian optimization for safe robot grasping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

[Rasmussen and Williams, 2006] Rasmussen, C. E. and Williams, C. K. (2006). *Gaussian processes for machine learning*, volume 1. MIT press Cambridge.

[Rolland et al., 2018] Rolland, P., Scarlett, J., Bogunovic, I., and Cevher, V. (2018). High-dimensional Bayesian optimization via additive models with overlapping groups. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 298–307.

[Ru et al., 2017] Ru, B., Osborne, M., and McLeod, M. (2017). Fast information-theoretic Bayesian optimisation. *arXiv preprint arXiv:1711.00673*.

[Scarlett et al., 2017] Scarlett, J., Bogunovic, I., and Cevher, V. (2017). Lower bounds on regret for noisy Gaussian process bandit optimization. In *Conference on Learning Theory (COLT)*.

[Shekhar and Javidi, 2018] Shekhar, S. and Javidi, T. (2018). Gaussian process bandits with adaptive discretization. *Electronic Journal of Statistics*, 12(2):3829–3874.

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

[Song et al., 2019] Song, J., Chen, Y., and Yue, Y. (2019). A general framework for multi-fidelity bayesian optimization with gaussian processes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

[Srinivas et al., 2010] Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. (2010). Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning (ICML)*, pages 1015–1022.

[Sui et al., 2015] Sui, Y., Gotovos, A., Burdick, J., and Krause, A. (2015). Safe exploration for optimization with Gaussian processes. In *International Conference on Machine Learning (ICML)*, pages 997–1005.

[Valko et al., 2013] Valko, M., Korda, N., Munos, R., Flaounas, I., and Cristianini, N. (2013). Finite-time analysis of kernelised contextual bandits. *Uncertainty In Artificial Intelligence (UAI)*.

[Vanchinathan et al., 2014] Vanchinathan, H. P., Nikolic, I., De Bona, F., and Krause, A. (2014). Explore-exploit in top-n recommender systems via Gaussian processes. In *ACM Conference on Recommender Systems*, pages 225–232.

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