# Instance-dependent Regret Analysis of Kernelized Bandits

**Shubhanshu Shekhar** [1] **Tara Javidi** [2]

## Abstract

We study the problem of designing an adaptive strategy for querying a noisy zeroth-order-oracle to efficiently learn about the optimizer of an unknown function $f$. To make the problem tractable, we assume that $f$ lies in the reproducing kernel Hilbert space (RKHS) associated with a known kernel $K$, with its norm bounded by $M < \infty$. Prior results, working in a *minimax framework*, have characterized the worst-case (over all functions in the problem class) limits on regret achievable by *any* algorithm, and have constructed algorithms with matching (modulo polylogarithmic factors) worst-case performance for the Matérn family of kernels. These results suffer from two drawbacks. First, the minimax lower bound gives limited information about the limits of regret achievable by the commonly used algorithms on a specific problem instance $f$. Second, the existing upper bound analysis fails to adapt to easier problem instances within the function class. Our work takes steps to address both these issues. First, we derive *instance-dependent* regret lower bounds for algorithms with uniformly (over the function class) vanishing normalized cumulative regret. Our result, valid for several practically relevant kernelized bandits algorithms, such as, `GP-UCB`, `GP-TS` and `SupKernelUCB`, identifies a fundamental complexity measure associated with every problem instance. We then address the second issue, by proposing a new minimax near-optimal algorithm which also adapts to easier problem instances.

## 1. Introduction

We consider the problem of optimizing a function $f : \mathcal{X} = [0, 1]^d \mapsto \mathbb{R}$ by adaptively gathering information about it via noisy zeroth-order-oracle queries. To make the problem tractable, we assume that **(1)** $f$ lies in the reproducing kernel Hilbert space (RKHS) associated with a given positive-definite kernel $K$, and **(2)** the RKHS norm of $f$ is bounded by a known constant $M < \infty$. The function $f$ can be accessed through noisy zeroth-order-oracle queries that return $y_x = f(x) + \eta$ at a query point $x \in \mathcal{X}$, with $\eta$ denoting the additive observation noise. Given a total budget $n$, the goal of an agent is to design an adaptive querying strategy, denoted by $\mathcal{A}$, to select a sequence of query points $(x_t)_{t=1}^n$, that incur a small cumulative regret $\mathcal{R}_n(\mathcal{A}, f)$, defined as

$$\mathcal{R}_n(\mathcal{A}, f) := \sum_{t=1}^{n} f(x^*) - f(x_t), \quad x^* \in \arg\max_{x \in \mathcal{X}} f(x).$$

The cumulative regret forces the agent to address the exploration-exploitation trade-off and prevents it from querying too many points from the sub-optimal regions of the domain.

The problem described above is referred to as the kernelized bandit or *agnostic* Gaussian Process bandit problem (Srinivas et al., 2012). Prior theoretical works in this area have focused on establishing lower and upper bounds on the performance achievable by algorithms in the minimax setting. These results characterize the worst-case limits, over all functions in the given RKHS, of performance achievable by any adaptive sampling algorithm (see Section 1.1 for details). The minimax framework does not account for the fact that there may exist functions that are easier to optimize than others in the same function class. As a result, the minimax regret bounds may not accurately reflect the performance of carefully designed adaptive strategies in the typical, non-adversarial, problem instances. In this paper, we take a step towards addressing this issue and present the first *instance-dependent* analysis for kernelized bandits.

In this paper, we focus on the RKHS corresponding to the Matérn family of kernels, denoted by $\{\mathcal{H}_{K_\nu} : \nu > 0\}$. These kernels are highly relevant for practical applications and allow a graded control over the smoothness of its elements (through a smoothness parameter $\nu > 0$) as described by Stein (2012). We believe that our arguments can be extended to Squared-Exponential (SE) kernels by using the Fourier domain constructions of Scarlett et al. (2017), al-

[1]Carnegie Mellon University [2]University of California, San Deigo. Correspondence to: Shubhanshu Shekhar <shubhan2@andrew.cmu.edu>.

*Proceedings of the $39^{th}$ International Conference on Machine Learning*, Baltimore, Maryland, USA, PMLR 162, 2022. 

though the intermediate steps may be more complicated. Furthermore, we also restrict our attention primarily to analyzing the cumulative regret, and leave the extension of our work to the pure exploration setting for future work (see Section 5 for further discussion).

The rest of this paper is organized as follows: we discuss the limitations of the existing minimax analysis and present an overview of our contributions in Section 1.1. We formally state the problem and the required assumptions in Section 2, and in Section 3 we derive the instance-dependent lower bounds of the regret achievable by a 'uniformly good' class of algorithms (this class includes all the existing algorithms analyzed in the literature including `GP-UCB`, `GP-TS`, `SupKernelUCB`). Finally, in Section 4.1, we propose a new algorithm that achieves the *best of both worlds*: it matches the minimax lower bounds (up to polylogarithmic factors) for the Matérn family of kernels in the worst case, and can also exploit some additional structure present in the given problem instance to achieve regret tighter than the minimax lower bound for those instances.

### 1.1. Overview of Results

For a given class of functions, $\mathcal{H}$, the minimax expected cumulative regret is defined as $\mathcal{R}_n^*(\mathcal{H}) := \inf_{\mathcal{A}} \sup_{f\in\mathcal{H}} \mathbb{E}\left[\mathcal{R}_n\left(f,\mathcal{A}\right)\right]$. For the RKHS associated with Matérn kernels with smoothness parameter $\nu > 0$, prior work has established a minimax rate $\mathcal{R}_n^*(\mathcal{H}_{K_\nu}) = \Theta\left(n^{(\nu+d)/(\nu+2d)}\right)$, ignoring polylogarithmic factors in $n$. In particular, the worst-case algorithm independent lower bound of the order $\Omega\left(n^{(\nu+d)/(2\nu+d)}\right)$ for the Matérn kernels with smoothness parameter $\nu > 0$ was established by Scarlett et al. (2017). On the other hand, Vakili et al. (2021a) recently derived tighter bounds on the mutual information gain (or equivalently the *effective dimension*) associated with Matérn kernels, that in turn implied that the `SupKernelUCB` algorithm of Valko et al. (2013) matches (up to polylogarithmic terms) the above-stated lower bound, hence showing its near-optimality.

While the existing theoretical results provide a rather complete understanding of the minimax regret for Matérn family of kernels, they suffer from two drawbacks:

1. The minimax lower bounds are obtained by constructing a suitable subset of 'hard' problem, and then showing that there exists no algorithm that can perform well on all of those problems simultaneously. These results tell us that for *any* algorithm, there exists at least one *hard problem instance* on which that algorithm must incur a certain regret. However, such results do not tell us what are the limits of performance for *carefully designed, 'good'* algorithms (such as `GP-UCB`; precise meaning of 'good' is stated in Definition 2) on *specific, non-adversarial, problem instances*.
2. The existing analysis of most of the algorithms depend on global properties of the given function class, such as the dimension, kernel parameters and RKHS norm. Consider, for example, the `GP-UCB` algorithm for which Srinivas et al. (2012) derived the following upper bound on the regret $\mathcal{R}_n = \widetilde{\mathcal{O}}\left(\sqrt{n\gamma_n}\right)$, where $\gamma_n$ is the maximum information gain for the given kernel $K$ formally defined in (2). Since $\gamma_n$ is a property associated with the kernel itself, the existing theory does not adapt to any simplifying structure that may be present in the specific problem instance.

Our main contributions make progress towards addressing the two issues stated above. In particular, we first derive an instance-dependent lower bound for algorithms with uniformly bounded normalized cumulative regret, and then we propose an algorithm that achieves near-optimal worst case performance but also can exploit some additional structure present in problem instances.

First, we consider the following question: *Suppose we are given an algorithm $\mathcal{A}$ that is known to have $\mathcal{O}\left(n^{a_0}\right)$ worst-case regret over all functions in the RKHS associated with Matérn kernel $K_\nu$ (denoted by $\mathcal{H}_{K_\nu}$). Then, what values of expected regret can $\mathcal{A}$ achieve for a given function $f$ in $\mathcal{H}_{K_\nu}$?* We answer this question by identifying a *lower-complexity* term $\underline{\mathcal{C}}_f$ that characterizes the per-instance achievable limit (precise statement in Theorem 1).

**Main Result 1** (Lower Bound)**.** *Suppose an algorithm $\mathcal{A}$ has a worst-case regret $\mathcal{O}\left(n^{a_0}\right)$ over functions in $\mathcal{H}_{K_\nu}$. Then, for a given $f$ with $\|f\|_{\mathcal{H}_{K_\nu}} < M$, we have:*

$$\mathbb{E}[\mathcal{R}_n(f,\mathcal{A})] = \Omega\left(\underline{\mathcal{C}}_f(n^{-(1-a_0)})\right),$$

*where $\underline{\mathcal{C}}_f(\Delta) := \sum_{k\geq 0} m_k/(2^{k+2}\Delta)$ for any $\Delta > 0$, and $m_k$ denotes the $2w_k = \mathcal{O}\left((\Delta 2^k)^{1/\nu}\right)$ packing number (see Definition 15 in Appendix A) of the annular set $\mathcal{Z}_k = \{x \in \mathcal{X} : 2^k\Delta \leq f(x^*) - f(x) < 2^{k+1}\Delta\}$.*

To interpret the term $\underline{\mathcal{C}}_f$, let us deconstruct each element, $m_k/(2^{k+2}\Delta)$, in its defining sum. Consider a ball (denoted by $E$) of radius $w_k = \mathcal{O}((2^k\Delta)^{1/\nu})$ contained in the region $\mathcal{Z}_k$. By construction of $\mathcal{Z}_k$, every point in $E$ is at most $2^{k+1}\Delta$ (and at least $2^k\Delta$) suboptimal for $f$. We first show that, if $\Delta > n^{-(1-a_0)}$, then $\mathcal{A}$ must spend at least $\Omega\left(1/(2^{k+1}\Delta)^2\right)$ queries in the region $E$. This implies that the total number of queries in the region $\mathcal{Z}_k$ is at least $\Omega(m_k/(2^{k+1}\Delta)^2)$, owing to the fact that $m_k$ disjoint balls of radius $w_k$ can be packed in the region $\mathcal{Z}_k$. This in turn, implies that the total regret incurred by these queries is lower bounded by $\Omega\big(2^k\Delta m_k/(2^{k+1}\Delta)^2\big) = \Omega\left(m_k/(2^{k+2}\Delta)\right)$. Further details of this argument are in Section 3 and in Appendix B.1.

The previous result can be specialized for minimax-optimal case by setting $a_0 = a^*_\nu := (\nu + d)/(2\nu + d)$, to obtain the per-instance performance limit of minimax optimal algorithms. Furthermore, with a careful choice of the problem instance $f$, we can also recover the minimax lower bound (see Remark 4). This motivates our next question: *Can we construct a minimax-optimal algorithm, that adapts, and incurs smaller regret on easier problem instances?* We address this question, by constructing a new algorithm $\mathcal{A}_1$ in Section 4.1, for which we show the following (see Theorem 2 for a more precise statement).

**Main Result 2** (Upper Bound)**.** *We construct an algorithm, $\mathcal{A}_1$, that is minimax near-optimal for functions in $\mathcal{H}_{K_\nu}$, and satisfies (with $a^*_\nu = (\nu + d)/(2\nu + d)$)*

$$\mathbb{E}[\mathcal{R}_n(f, \mathcal{A}_1)] = \widetilde{\mathcal{O}}\left(\overline{\mathcal{C}}_f(n^{-(1-a^*_\nu)})\right),$$

*where $\overline{\mathcal{C}}_f(\Delta) := \sum_{k\geq 0} \widetilde{m}_k/(2^{k\xi}\Delta)$ for $\Delta > 0$ and $\xi = \min\{1, \nu\}$. The term $\widetilde{m}_k$ is the $2\widetilde{w}_k = \mathcal{O}\big((2^{k\xi}\Delta)^{1/\xi}\big)$ packing number of the set $\widetilde{\mathcal{Z}}_k = \{x : f(x^*) - f(x) = \mathcal{O}(2^{k\xi}\Delta)\}$.*

Similar to $\underline{\mathcal{C}}_f$, the upper-complexity term $\overline{\mathcal{C}}_f$ can also be interpreted in terms of the number of queries made by our proposed algorithm $\mathcal{A}_1$ in the regions $\widetilde{\mathcal{Z}}_k$. In general, the term $\overline{\mathcal{C}}_f$ is larger than $\underline{\mathcal{C}}_f$. This is mainly due to two reasons:

- We have stated our lower bound result for algorithms with a worst-case $\mathcal{O}(n^{a_0})$ regret over the entire function class $\mathcal{H}_{K_\nu}$. However, the proof only requires the regret guarantee to hold uniformly for a small class of local perturbations. Thus, it is possible to refine the result by imposing a uniform regret condition only in a small neighborhood of $f$ (with radius shrinking with $n$ at an appropriate rate). We leave the investigation of such refined lower bounds to future work.
- The packing radius $\widetilde{w}_k$ used in $\overline{\mathcal{C}}_f$ is smaller than the analogous term $w_k$ used in $\underline{\mathcal{C}}_f$, due to the presence of $\xi = \min\{1, \nu\}$ instead of $\nu$.

Despite these issues, in Proposition 2 we identify conditions under which $\overline{\mathcal{C}}_f$ is strictly tighter than the minimax rate, thus proving that $\mathcal{A}_1$ can adapt to easier problem instances.

In the case of functions where the abstract packing numbers $m_k$ and $\widetilde{m}_k$ can be well-estimated, the above results give us explicit (in $n$) regret bounds. For example, if the function $f \approx \|x - x^*\|^b$ for some $b > \nu$ in the neighborhood of its maximizer $x^*$ (formally stated in Definition 4), then we can obtain explicit instance dependent lower (Proposition 1) and upper (Proposition 2) bound. In Figure 1, we plot the variation of the exponent of the regret bound (i.e., $\alpha$ if regret is $\approx n^\alpha$) with dimension for the existing upper bounds for different algorithms (dashed) and their corresponding instance-dependent lower bounds (solid-lines, same color) as well as the instance-dependent upper bound of our algorithm $\mathcal{A}_1$ for $\nu = 1.1$ and $b = 1.2$.

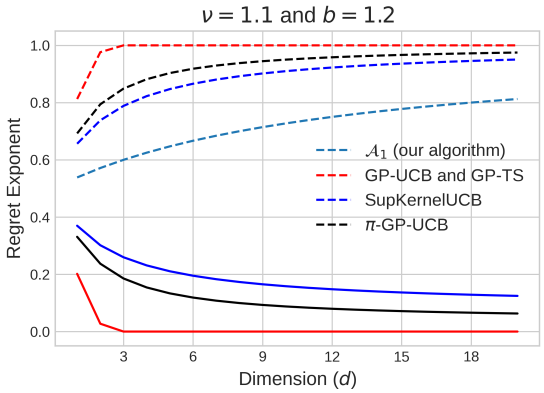


*Figure 1.* Plot of the variation of the exponent of different regret upper and lower bounds (i.e., $\alpha$, if regret $\approx n^\alpha$) with dimension, on an instance $f \in \mathcal{H}_{K_\nu}$, satisfying $f \approx (x - x^*)^b$ near its optimum. This "local growth" property of $f$, formally stated in Definition 4, allows us to obtain closed form estimates of $\underline{\mathcal{C}}_f$ and $\overline{\mathcal{C}}_f$. The dashed lines denote upper bounds, and the solid lines of the same color represent the corresponding instance-dependent lower bounds derived in Proposition 1. Note that the solid blue line represents the instance dependent lower bound for both, $\mathcal{A}_1$ and `SupKernelUCB`, as they are both minimax near-optimal.

### 1.2. Related Work

*Minimax lower bounds for kernelized bandits.* Scarlett et al. (2017) derived the first algorithm-independent minimax lower bound on the regret for the kernelized bandits problem with SE and Matérn kernels. For Matérn kernel $K_\nu$, they obtain an algorithm-independent minimax lower bound of $\Omega\left(n^{(d+\nu)/(d+2\nu)}\right)$. Their result proceeds by first constructing a finite class of *hard* problem instances, and then using a reduction to multiple hypothesis testing and a change of measure argument. However, the *hard* functions employed by Scarlett et al. (2017) are of the *needle-in-haystack* type, and may not be representative of the typical functions belonging to the RKHS. Thus, the resulting regret lower bound may provide a pessimistic limit for the achievable performance of algorithms on the specific problem instance encountered.

*Instance-dependent lower bounds for linear bandits.* Instance dependent lower bounds have been derived for the related problem of linear bandits with finitely many arms, either in the pure-exploration (Degenne et al., 2020) or regret-minimization (Tirinzoni et al., 2020) settings. At a high level, the proofs of these results, as well as our Theorem 1, proceed in two steps: (i) given a problem instance,

identify a suitable class of alternative problem instances, and (ii) obtain a lower bound on the number of queries that any *uniformly good* algorithm must allocate to the alternatives. For the case of finite domains considered in the above-mentioned works, the alternative classes are quite naturally defined in terms of instances with different optimal arm. This is not the case in our setting due to the continuous nature of the domain $\mathcal{X} = [0,1]^d$. We address this by presenting a perturbation argument to construct suitable alternatives depending on the problem instance as well as the budget $n$ to derive our instance-dependent lower bounds.

*Kernelized bandit algorithms.* Srinivas et al. (2012) proposed the GP−UCB algorithm motivated by the Upper Confidence Bound (UCB) strategy for multi-armed bandits (MABs) (Auer et al., 2002), which proceeds by selecting query points $(x_t)_{t\geq 1}$ that maximize the UCB of $f$ of the form $\mu_t(x) + \beta_t \sigma(t)(x)$ over a sequence of finite discretizations of the domain $\mathcal{X}$ for suitable factors $(\beta_t)_{t\geq 1}$. For this algorithm, Srinivas et al. (2012) derived the following high-probability upper bound on $\mathcal{R}_n$

$$\mathcal{R}_n = \tilde{\mathcal{O}}\left(\sqrt{n\gamma_n}\right). \tag{1}$$

In the above display $\gamma_n$ is the maximum information gain associated with the kernel $K$, defined as

$$\gamma_n \coloneqq \max_{S\subset\mathcal{X}:|S|=n} I(\boldsymbol{y}_S;\, f), \quad \text{for} \quad f \sim GP(0,K), \tag{2}$$

where $\boldsymbol{y}_S = (y_1,\ldots,y_n)$ is the vector of observations at points in $S = (x_1,\ldots,x_n)$ and $I(\boldsymbol{y}_S; f)$ denotes the mutual information between the observations $\boldsymbol{y}_S$ and the function $f$ assumed to be a sample from a zero-mean Gaussian Process $GP(0,K)$. To obtain explicit (in $n$) regret bounds from (1), Srinivas et al. (2012) also derived upper bounds on $\gamma_n$ for the SE and Matérn families. More recently, Vakili et al. (2021a) derived tighter bounds on $\gamma_n$ for these families using a different approach than that employed by Srinivas et al. (2012). In particular, for the Matérn family, this implies the following regret bound for GP−UCB: $\mathcal{R}_n = \tilde{\mathcal{O}}\left(n^{(3d/2+\nu)/(d+2\nu)}\right)$ where $\nu$ is the smoothness parameter. Chowdhury and Gopalan (2017) derived that the same upper bound for the Thompson Sampling based randomized algorithm, GP−TS.

The regret bounds of GP−UCB and GP−TS for Matérn kernels, are not sublinear when $\nu < d/2$. Janz et al. (2020) addressed this issue by proposing an algorithm (referred to as $\pi$−GP−UCB), which adaptively partitions the input space and fits independent GP models in each element of the partition. This yields an alternative bound on $\gamma_n$, and results in a sublinear regret bound of the form $\mathcal{R}_n = \tilde{\mathcal{O}}\left(n^{e_\nu}\right)$, where $e_\nu = {}^{d(2d+3)+2\nu}/_{d(2d+4)+4\nu}$, for $\nu > 1$ and $d \geq 1$.

Valko et al. (2013) proposed the SupKernelUCB algorithm that proceeds by dividing the queried points into batches which consist of conditionally independent observations. Although initially proposed for finite domains, the algorithm easily extends to continuous domains by using sufficiently fine discretizations. Valko et al. (2013) showed that the SupKernelUCB algorithm achieves a regret bound of $\mathcal{R}_n = \tilde{\mathcal{O}}\left(\sqrt{n\gamma_n}\right)$; tighter than the bound for GP−UCB (and GP−TS) by a factor of $\sqrt{\gamma_n}$. By plugging in the recently derived bounds on $\gamma_n$ for Matérn kernels by Vakili et al. (2021a), this implies that the SupKernelUCB algorithm achieves a regret bound $\mathcal{R}_n = \tilde{\mathcal{O}}\left(n^{(d+\nu)/(d+2\nu)}\right)$, matching the minimax lower bounds.

Despite being minimax near-optimal, SupKernelUCB often performs poorly in practice. Recent works, such as (Salgia et al., 2021; Li and Scarlett, 2022), address this by designing algorithms that are minimax near-optimal, and also perform well on common benchmarks. Finally, we note that another minimax near-optimal kernelized bandit algorithm was recently proposed by Camilleri et al. (2021) using a sampling strategy based on G-optimal designs.

## 2. Preliminaries

In this section, we introduce some definitions and notations required to formally state the problem. Additional definitions are deferred to Appendix A.

As stated in the introduction, we consider the problem of optimizing a black-box function $f : \mathcal{X} \mapsto \mathbb{R}$, where $\mathcal{X} = [0,1]^d$ that can be accessed through noisy zeroth order oracle queries. The task of the learner (or agent) is to design an adaptive strategy to select query points $\{x_t : 1 \leq t \leq n\}$, that incur a small cumulative regret $\mathcal{R}_n(f,\mathcal{A}) = \sum_{t=1}^n f(x^*) - f(x_t)$. To formally present the problem statement, we need a precise definition of adaptive querying strategies and the exact assumptions made on the objective function and observation noise.

We begin with a definition of adaptive querying strategy.

**Definition 1** (Adaptive Strategy)**.** An adaptive querying strategy $\mathcal{A}$ consists of a sequence of mappings $(A_t)_{t=1}^\infty$ where $A_t : (\mathcal{X}\times\mathcal{Y})^{t-1} \times \mathcal{U}^t \mapsto \mathcal{X}$, where $\mathcal{U} = [0,1]$ represents the range of additional randomness (used in randomized algorithms such as GP−TS).

For the rest of the paper, we will use the terms adaptive strategy and algorithm interchangeably. Next, we introduce the assumptions on the objective function and observation noise under which our theoretical results will be stated.

**Assumption 1.** The objective function $f$ lies in the RKHS $\mathcal{H}_K$ associated with a known kernel $K$, and its RKHS norm $\|f\|_{\mathcal{H}_K}$ is bounded by a known constant $M < \infty$.

The above assumption is standard in the kernelized bandits literature, and informally it states that the unknown function has *low complexity*, where the complexity is quantified in

terms of the RKHS norm. As stated in the introduction, we will restrict our discussion to Matérn kernels in this paper (formally defined in Appendix A).

**Assumption 2a.** The zeroth-order-oracle returns $y(x_t) = f(x_t) + \eta_t$ for a query point $x_t \in \mathcal{X}$, where $\{\eta_t : t \geq 1\}$ is an i.i.d. sequence of $N(0, \sigma^2)$ random variables.

**Assumption 2b.** The zeroth-order-oracle returns $y(x_t) = f(x_t) + \eta_t$ for a query point $x_t \in \mathcal{X}$, where $\{\eta_t : t \geq 1\}$ is an i.i.d. sequence of $\sigma^2$-sub-Gaussian random variables.

These two assumptions state that the observation noise has light tails, and hence we can construct tight confidence intervals for the unknown functions based on the noisy observations. We will use Assumption 2a in the statement of our lower bound, while the more general Assumption 2b will be used to state the upper bound result. Note that imposing the $N(0, \sigma^2)$ requirement for stating the lower bounds is primarily to simplify the presentation. This is further discussed in Remark 9 in Appendix B.2.1, and relies on the fact that we can obtain closed form expressions for KL-divergence involving Gaussian random variables, which simplifies the expression of the lower-complexity term (see Definition 3).

We now present the formal problem statement.

**Problem Statement 1.** *Suppose Assumptions 1 and either 2a or 2b hold with known values of $M$ and $\sigma^2$. Then, given a total querying budget $n$, design an adaptive strategy $\mathcal{A}$ to select query points $x_1, \ldots, x_n$, which incur a small cumulative regret $\mathcal{R}_n(\mathcal{A}, f) \coloneqq \sum_{t=1}^{n} f(x^*) - f(x_t)$.*

*Notations.* We end this section, by listing some of the notations that will be used in the rest of the paper. As mentioned earlier, $\mathcal{X} = [0,1]^d$ for some $d \geq 1$ represents the domain and $f : \mathcal{X} \mapsto \mathbb{R}$ is the unknown objective function. We will use $x^*$ to represent an optimal point of $f$, i.e., $x^* \in \arg\max_{x \in \mathcal{X}} f(x)$. For any $x \in \mathcal{X}$ and $r > 0$, we use $B(x, r)$ to denote the $\ell_2$ open ball $\{z \in \mathcal{X} : \|z - x\|_2 < r\}$ and $B(x, r_1, r_2)$ to denote the annular region $\{z \in \mathcal{X} : r_1 < \|z - x\| < r_2\}$. For a function $f$ and constants $u > 0$ and $c > 1$, we use $\widetilde{\mathcal{X}}(f, u, c)$ to denote the set $\{x \in \mathcal{X} : u \leq f(x^*) - f(x) \leq cu\}$.

## 3. Lower Bounds

In order to derive instance-dependent bounds on the regret, we need to restrict our attention to 'uniformly-good' algorithms that perform well for all elements of the given problem class. A restriction of this kind is necessary to obtain non-trivial instance-dependent regret bounds. Otherwise, for every problem instance $f$, there exists a trivial algorithm that always queries $x^* \in \arg\max_{x \in \mathcal{X}} f(x)$, and incurs no regret on $f$, but suffers a linear regret for all functions for which $x^*$ is strictly suboptimal.

We begin by presenting a definition of $a_0$-consistent algorithms that formally characterize the meaning of 'uniformly good' algorithms required for obtaining instance-dependent lower bound. This definition is motivated by a similar notion of consistent policies used in multi-armed bandits (see Lattimore and Szepesvári, 2020, Definition 16.1).

**Definition 2** ($a_0$-consistency)**.** An algorithm $\mathcal{A}$ is said to be $a_0$-consistent over a function class $\mathcal{F}$, if for all $a > a_0$ and $f \in \mathcal{F}$, the following holds:

$$\lim_{n \to \infty} \frac{\mathbb{E}\left[\mathcal{R}_n(\mathcal{A}, f)\right]}{n^a} = 0.$$

*Remark* 1. Note that when $\mathcal{F}$ is the RKHS associated with a Matérn kernel, then all the existing algorithms discussed in Section 1.2 satisfy the condition above with some $a_0 < 1$. In particular, this condition is satisfied by `GP-UCB` and `GP-TS` with $a_0 = \min\{1, (\nu + 3d/2)/(2\nu + d)\}$, by $\pi$−`GP-UCB` with $a_0 = (d(2d+3) + 2\nu)/(d(2d+4) + 4\nu)$ and by `SupKernelUCB` with $a_0 = (\nu + d)/(2\nu + d)$.

The rest of this section is organized as follows. In Section 3.1, we first present a general lower bound that characterizes the regret achievable by an algorithm on a given function $f$, in terms of the lower complexity term $\underline{\mathcal{C}}_f$ introduced in Definition 3. This abstract complexity term depends on the packing number of certain near-optimal regions associated with $f$, as well as the exponent $a_0$ of the uniform regret condition for $a_0$−consistent algorithms. In Section 3.2, we consider the special case of functions satisfying a *local growth* condition (Definition 4), for which the term $\underline{\mathcal{C}}_f$ can be well-estimated to get an explicit lower bound in terms of the query budget $n$.

### 3.1. General lower bound

We now obtain a general instance-dependent lower bound for $a_0$-consistent algorithms. First, we introduce a notion of complexity associated with a function $f \in \mathcal{H}_{K_\nu}(M)$, that will be used to state the main result.

**Definition 3** (Lower-Complexity)**.** Let $f \in \mathcal{H}_{K_\nu}(M)$ with $\|f\|_{K_\nu} = (1-\lambda)M$ for some $\nu > 0$, $M > 0$ and $\lambda \in (0,1)$. Fix $\Delta > 0$, and introduce the set $\mathcal{Z}_k \coloneqq \widetilde{\mathcal{X}}(f, 2^k\Delta, 2) = \{x \in \mathcal{X} : 2^k\Delta \leq f(x^*) - f(x) < 2^{k+1}\Delta\}$. Introduce the radius $w_k = \left(3 \times 2^k \Delta M_\nu / (\lambda M)\right)^{1/\nu}$ (with $M_\nu$ denoting the norm of the bump function introduced in Definition 17 in Appendix B.1.2) and let $m_k$ denote the $2w_k$-packing number of the set $\mathcal{Z}_k$. Finally, define the complexity term

$$\underline{\mathcal{C}}_f(\Delta) \coloneqq \sum_{k \geq 0} \frac{m_k}{2^{k+2}\Delta} \geq \frac{m_0}{4\Delta}. \tag{3}$$

Note that our notation $\underline{\mathcal{C}}_f(\Delta)$ suppresses the $\nu, M$ and $\lambda$ dependence of $\underline{\mathcal{C}}_f$.

We now present an instance-dependent lower bound on the expected cumulative regret of any $a_0$-consistent algorithm $\mathcal{A}$ in terms of the complexity term introduced above.

**Theorem 1.** *Let $f \in \mathcal{H}_{K_\nu}(M)$ with $\|f\|_{K_\nu} = (1-\lambda)M$ for $\nu > 0, M > 0$ and $\lambda \in (0,1)$. Consider any $a_0$-consistent algorithm (with $a_0 < 1$) over the family of functions $\mathcal{H}_{K_\nu}(M)$, denoted by $\mathcal{A}$, and fix any $a > a_0$. Then, the expected cumulative regret of $\mathcal{A}$ on the instance $f$ satisfies*

$$\mathbb{E}\left[\mathcal{R}_n\left(\mathcal{A}, f\right)\right] \geq \frac{7\log 2}{4}\sigma^2 \underline{\mathcal{C}}_f\left(n^{-(1-a)}\right),$$

*for $n$ large enough (exact condition in equation 13 in Appendix B).*

Theorem 1 follows as a consequence of a more general statement, presented and proved in Appendix B. We also present a detailed overview of the argument involved in proving Theorem 1 in Appendix B.1. The high level idea is to first construct a perturbed version of $f$ by adding a bump function (see Definition 17 in Appendix B.1.2) supported on a ball $E$ that is contained in a region $\mathcal{Z}_k$ for some $k \geq 1$ (see Definition 3). Then, using the $a_0$-consistency of $\mathcal{A}$, we obtain a lower bound on the number of times $\mathcal{A}$ must query the region $E$. By identifying the total number of disjoint balls that can be packed in different regions of the input space, and adding their contributions to the regret, we obtain the final result.

*Remark* 2. The lower bound in Theorem 1 based on the complexity term of eq. (3) has a natural interpretation. For any $k \geq 0$, the set $\mathcal{Z}_k$ denotes the 'annular' region where the suboptimality $f(x^*) - f(x)$ is between $2^k\Delta_n$ and $2^{k+1}\Delta_n$, with $\Delta_n := n^{-(1-a)}$. Then, the term $w_k = \left(2^{k+1}\Delta_n/(\lambda M)\right)^{1/\nu}$ denotes the radius of the smallest ball that can support a scaled bump function (see Definition 17) that ensures the resulting perturbed version of $f$ still remains in $\mathcal{H}_{K_\nu}(M)$ and also satisfies the properties stated in Definition 16 in Appendix B.1.1. As we show in Appendix B.1.1, for any such perturbation of $f$, the algorithm $\mathcal{A}$ must spend roughly $1/(2^{k+1}\Delta_n)^2$ samples to distinguish between $f$ and its perturbation. The regret incurred in the process is lower bounded by $2^k\Delta_n \times \left(1/(2^{k+1}\Delta_n)^2\right) = 1/(2^{k+2}\Delta_n)$, that is, suboptimality ($\geq 2^k\Delta_n$) times the number of queries in that region ($\geq 1/(2^{k+1}\Delta_n)^2$). Since $m_k$ disjoint balls of radius $w_k$ can be packed into $\mathcal{Z}_k$, the expression of the complexity term in (3) follows.

The strict inequality in the definition of the complexity term in (3) immediately implies the following weaker, but more interpretable, version of the above statement.

**Corollary 1.** *Consider an $a_0$-consistent algorithm $\mathcal{A}$, and fix any $a > a_0$. Under the same assumptions as Theorem 1, and with $w_0 := \left(3(n^{-(1-a)})/(\lambda M)\right)^{1/\nu}$, let $m_0$ denote the $2w_0$-packing number of the set $\mathcal{Z}_0 := \{x \in \mathcal{X} : n^{-(1-a)} \leq f(x^*) - f(x) < 2n^{-(1-a)}\}$. Then, we have the following:*

$$\mathbb{E}\left[\mathcal{R}_n\left(f, \mathcal{A}\right)\right] = \Omega\left(\sigma^2 m_0\, n^{(1-a)}\right).$$

*Remark* 3. Corollary 1 states that a key quantity characterizing the regret achievable by a uniformly good algorithm is the packing number of an annular near-optimal region associated with the given function. Informally, we can write $m_0 \approx w_0^{-\tilde{d}} \approx n^{(1-a)\tilde{d}/\nu}$, where $\tilde{d} := \liminf_{n\to\infty} \log(m_0)/\log(1/w_0)$. The term $\tilde{d}$ is reminiscent of the concepts of *near-optimality dimension* and *zooming dimension* used in prior works in bandits in metric spaces (Bubeck et al., 2011; Kleinberg et al., 2019) as well as in Gaussian Process bandits (Shekhar and Javidi, 2018). These works use such notions to obtain instance-dependent upper bounds on the regret of algorithms that non-uniformly discretize the domain. This connection also motivates our approach in designing Algorithm 1 in Section 4.1.

*Remark* 4. As a sanity check, we note that Corollary 1 recovers the minimax lower bound by considering $f$ to be a scaled bump function $g$ introduced in Definition 17, and $\mathcal{A}$ to be a minimax near-optimal algorithm. In particular, by defining $f(\cdot) = 2\Delta_n g(\cdot/w_0)$, the region $\mathcal{Z}_0$ contains the whole domain with a ball of radius $w_0$ removed. This implies that $m_0 \approx w_0^{-d} = n^{(1-a)d/\nu}$ for $a > a^*_\nu = (d+\nu)/(d+2\nu)$, and thus $\mathbb{E}[\mathcal{R}_n(f,\mathcal{A})] = \Omega\left(n^\alpha\right)$ for any $\alpha < (1-a^*_\nu)\left(1+\frac{d}{\nu}\right) = \frac{\nu}{d+2\nu} \times \frac{d+\nu}{\nu} = \frac{d+\nu}{d+2\nu} = a^*_\nu$.

In the next section, we specialize the above results to functions that satisfy an additional 'local growth' condition, for which the complexity terms can be explicitly lower bounded in terms of more interpretable parameters.

### 3.2. Lower bound under growth condition

In this section, we consider the class of functions satisfying the following additional assumption.

**Definition 4** (Growth Condition)**.** We say that the objective function $f$ satisfies the local growth condition with parameters $(\underline{c}, \bar{c}, b, \varrho_0)$ if for all $x \in B(x^*, \varrho_0) \cap \mathcal{X}$, we have $\underline{c}\|x - x^*\|^b \leq f(x^*) - f(x) \leq \bar{c}\|x - x^*\|^b$. We shall denote by $\mathcal{F}\left(\underline{c}, \bar{c}, b, \varrho_0\right)$ the class of all functions satisfying this property.

Similar conditions have been used in analyzing the performance of first order stochastic optimization algorithms by Ramdas and Singh (2013) and in characterizing the minimax rates of active learning algorithms by Castro and Nowak (2008).

As an example, consider the case when the function $f$ has continuous second order derivatives and its optimizer $x^*$ lies in the interior of the domain. Then, if the Hessian of $f$ at $x^*$ is non-singular, we can find a $\varrho_0 > 0$ such that for all $x$ in $B(x^*, \varrho_0)$, the spectral norm of the Hessian of $f$ is between $\underline{c}$ and $\bar{c}$. Then the function $f$ satisfies the growth condition with exponent 2 and constants $\underline{c}$ and $\bar{c}$ depending on the minimum and maximum norm of the Hessian in $B(x^*, \varrho_0)$. We now state the main result of this section.

**Proposition 1.** *Introduce the function class* $\mathcal{G} := \mathcal{H}_{K_\nu}(M) \cap \mathcal{F}(\underline{c}, \bar{c}, b, \varrho_0)$ *for some* $\nu > 0$ *and* $b > \nu$. *Let* $\mathcal{A}$ *be an* $a_0$*-consistent algorithm with* $a_0 < 1$ *for the function class* $\mathcal{H}_{K_\nu}(M)$. *Then for any* $a > a_0$ *and* $f \in \mathcal{G}$ *with* $\|f\|_{\mathcal{H}_{K_\nu}} < M$, *we have the following:*

$$\liminf_{n\to\infty} \frac{\mathbb{E}\left[\mathcal{R}_n(\mathcal{A}, f)\right]}{n^\alpha} > 0 \text{ for any } \alpha < (1-a_0)\left(1+\frac{d}{\nu}\left(1-\frac{\nu}{b}\right)\right)$$

Since most of the commonly used algorithms, such as `GP-UCB`, `GP-TS`, `SupKernelUCB` and $\pi-$`GP-UCB`, discussed further in Section 1.2, satisfy $a_0$-consistency introduced in Definition 2, we can use Theorem 1 to obtain the instance-dependent lower bounds for these algorithms.

**Corollary 2.** *Proposition 1 implies that* $\mathbb{E}[\mathcal{R}_n(\mathcal{A}, f)] = \Omega(n^\alpha)$, *when* $f \in \mathcal{G}$, $b > \nu$ *and*

- $\mathcal{A}$ *is the* `SupKernelUCB` *algorithm,* $\nu > 1$ *and* $\alpha < \left(\frac{\nu+d(1-\nu/b)}{2\nu+d}\right)$, *or*
- $\mathcal{A}$ *is either* `GP-UCB` *or* `GP-TS`, $\nu > d/2$, *and* $\alpha < \left(1-\frac{d}{2\nu}\right)\frac{\nu+d(1-\nu/b)}{2\nu+d}$, *or*
- $\mathcal{A}$ *is the* $\pi-$`GP-UCB` *algorithm,* $\nu > 1$ *and* $\alpha < \left(1+\frac{d}{2\nu}\right)\frac{\nu+d(1-\nu/b)}{2\nu+d(d+2)}$.

The results of the above corollary are presented for some specific $\nu = 1.1$ and $b = 1.2$ in Figure 1. As we can see, algorithms with tighter uniform regret bound (i.e., smaller $a_0$) incur higher instance-dependent lower bounds.

## 4. Instance-Dependent Upper Bound

As mentioned in the introduction, the prior theoretical analysis of the existing kernelized bandits algorithms upper bound their regret in terms of quantities such as the maximum information gain, $\gamma_n$, defined in (2). Since $\gamma_n$ is a property associated with the kernel $K$, such results do not adapt to the hardness of the specific problem instance within the associated RKHS – they predict the same performance guarantees for the 'easiest' as well as the 'hardest' problem in the class. As a result, the ability of the existing algorithms to adapt to the complexity of problems with the same problem class is unknown.

Our results in this section take a step towards addressing this issue. In particular, we describe a new algorithm for kernelized bandits in Section 4.1, and show in Section 4.2 that this algorithm is minimax near-optimal and also admits tighter upper bounds for easier problem instances.

### 4.1. Proposed Algorithm

We first introduce a standard notion of a sequence of nested partitions of the input space, often used in prior works, such as (Bubeck et al., 2011; Munos, 2011; Wang et al., 2014; Shekhar and Javidi, 2018), to design algorithms for zeroth order optimization.

**Algorithm 1:** Breadthwise Exploration with Adaptive Discretization ($\mathcal{A}_1$)

**Input:** $n$, the querying budget
$K_\nu$, the kernel belonging to the Matérn family
$M$, upper bound on the RKHS norm
$v_1, v_2, \rho$, parameters of the tree of partitions
$\tau$, regularization parameter used in posterior computation.

**Initialize:** $t = 1$, `flag` $\leftarrow True$, $\mathcal{E}_t \leftarrow \emptyset$, $\mathcal{Y}_t \leftarrow \emptyset$, $\mathcal{P}_t \leftarrow \{x_{0,1}\}$, $L = M \log n$ and $\xi = \min\{\nu, 1\}$.
**for** $t = 1, 2, \ldots, n$ **do**
  **while** `flag` **do**
    $\beta_t, \{(\mu_t(x), \sigma_t(x)) : x \in \mathcal{P}_t\} =$ `ComputePosterior`$(\mathcal{P}_t, \mathcal{E}_t, \mathcal{Y}_t, \tau)$
    $x_t \in \arg\max_{x\in\mathcal{P}_t} \sigma_t(x)$, $U_t = \max_{x\in\mathcal{P}_t} \sigma_t(x)$
    **if** $1/\sqrt{n} < U_t < L\left(v_1\rho^{h_t}\right)^\xi$ **then**
      $\mathcal{P}_t, h_t \leftarrow$ `RefinePartition`$(\mathcal{P}_t, \beta_t, \{(\mu_t(x), \sigma_t(x)) : x \in \mathcal{P}_t\})$
      $\mathcal{E}_t \leftarrow \emptyset$, $\mathcal{Y}_t \leftarrow \emptyset$
    **else**
      $y_t \leftarrow f(x_t) + \eta_t$, $\mathcal{E}_t \leftarrow \mathcal{E}_t \cup \{x_t\}$, $\mathcal{Y}_t \leftarrow \mathcal{Y}_t \cup \{y_t\}$;
      `flag` $\leftarrow False$
    **end**
  **end**
  `flag` $\leftarrow True$;
**end**

**Definition 5** (tree of partitions)**.** We say that a sequence of subsets of $\mathcal{X}$, denoted by $(\mathcal{X}_h)_{h\geq 0}$, forms a tree of partitions of $\mathcal{X}$, if it satisfies the following properties:

- For all $h \geq 0$, we have $\mathcal{X}_h = \{x_{h,i} : 1 \leq i \leq 2^h\}$. Furthermore, for every $x_{h,i} \in \mathcal{X}_h$ is associated a cell $\mathcal{X}_{h,i}$. For $i \neq j$, $\mathcal{X}_{h,i}$ and $\mathcal{X}_{h,j}$ are disjoint, and $\cup_{i=1}^{2^h}\mathcal{X}_{h,i} = \mathcal{X}$. Moreover, for any $h, i$ we have $\{x_{h+1,2i-1}, x_{h+1,2i}\} \in \mathcal{X}_{h+1} \cap \mathcal{X}_{h,i}$.
- There exist constants $0 < v_2 \leq 1 \leq v_1$ and $\rho \in (0,1)$ such that

$$B(x_{h,i}, v_2\rho^h) \subset \mathcal{X}_{h,i} \subset B(x_{h,i}, v_1\rho^h).$$

Next, we introduce two subroutines that will be employed by the algorithm. The first subroutine, called `ComputePosterior`, computes the posterior mean and posterior covariance function of the surrogate

Gaussian Process (GP) model used for approximating the unknown objective function in the algorithm.

**Definition 6** (`ComputePosterior`)**.** Given a subset $\mathcal{P}_t \subset \mathcal{X}$, a multi-set of points belonging to $\mathcal{P}_t$, denoted by $\mathcal{E}_t = \{x_1, \ldots, x_{t-1}\}$, at which the function $f$ was evaluated, the corresponding noisy function evaluations $\mathcal{Y}_t = \{y_1, \ldots, y_{t-1}\}$ and a constant $\tau > 0$, the `ComputePosterior` subroutine returns the terms $\beta_t$, and $\{(\mu_t(x), \sigma_t(x)) : x \in \mathcal{P}_t\}$, which are defined as follows:

$$\Sigma_t = K_t + \tau I, \qquad \text{where} \quad K_t = [K(x_i, x_j)]_{x_i, x_j \in \mathcal{E}_t}$$
$$k_t(x) = [K(x, x_1), \ldots, K(x, x_{t-1})]^T, \ \boldsymbol{y_t} = [y_1, \ldots, y_{t-1}]^T$$
$$\beta_t = \sqrt{2 \log \left( |\mathcal{P}_t| n^3 \right) / t}$$
$$\mu_t(x) = k_t(x)^T \Sigma_t^{-1} \boldsymbol{y_t}$$
$$\sigma_t(x) = \tau^{-1/2} \sqrt{K(x, x) - k_t(x)^T \Sigma_t^{-1} k_t(x)}.$$

As we describe later, our algorithm proceeds by adaptively discretizing the input space based on the observations gathered – the granularity of the partition becoming finer in the near-optimal regions of the input space, aimed at mimicking the discretization of the space involved in defining the complexity term in Definition 3. Our second subroutine, called `RefinePartition`, presents the formal steps involved in updating the discretization used in the algorithm.

**Definition 7** (`RefinePartition`)**.** This subroutine takes in as inputs, $\mathcal{P}_t, h_t, \beta_t$ and $\{(\mu_t(x), \sigma_t(x)) : x \in \mathcal{P}_t\}$; and returns an updated partition $\mathcal{P}_t$ and level $h_t$. First compute $l_t \coloneqq \max_{x \in \mathcal{P}_t} \mu_t(x) - \beta_t \sigma_t(x)$ and define $\tilde{\mathcal{P}}_t = \{x \in \mathcal{P}_t : \mu_t(x) + \beta_t \sigma_t(x) > l_t\}$. Next, define the new $\mathcal{P}_t$ as $\mathcal{P}_t = \cup_{x_{h_t,i} \in \tilde{\mathcal{P}}_t} \{x_{h_t+1,2i-1}, x_{h_t+1,2i}\}$, and update $h_t \leftarrow h_t + 1$.

We now present an outline of the steps of our proposed algorithm below. The formal pseudocode is in Algorithm 1.

*Outline of Algorithm 1.* At any time $t \geq 1$, the algorithm maintains a set of *active points* denoted by $\mathcal{P}_t$. These points satisfy the following two properties: **(i)** $\mathcal{P}_t \subset \mathcal{X}_{h_t}$ for some $h_t \geq 0$; i.e., all the active points lie in the same 'depth' (i.e., $h_t$) of the tree of partitions, and **(ii)** any optimizer $x^*$ of $f$ must lie in the region $\cup_{x_{h_t,i} \in \mathcal{P}_t} \mathcal{X}_{h_t,i}$. The algorithm evaluates the function at points in the active set, and computes the posterior mean and standard deviation by calling the `ComputePosterior` subroutine. The algorithm then compares the maximum posterior standard deviation (for points in $\mathcal{P}_t$) with an upper bound on the variation in function value in the cell $\mathcal{X}_{h_t,i}$ associated with an active point $x_{h_t,i} \in \mathcal{P}_t$. If the maximum posterior standard deviation is larger than $L(v_1 \rho^h)^\xi$, then the algorithm evaluates the function at the corresponding active point with the largest $\sigma_t(x)$. Otherwise, it concludes that the active points in $\mathcal{P}_t$ have been sufficiently well explored, and it moves to the next level of the partition tree by calling the `RefinePartition` subroutine. The above process continues until the querying budget is exhausted.

*Remark* 5. Algorithm 1 carefully combines two key ideas from kernelized bandits literature: **(i)** it divides the evaluated points into subsets (according to their level $h$ in the tree) which satisfy a conditional independence property, similar to `SupKernelUCB` of Valko et al. (2013), and **(ii)** adaptively partitions the input space to zoom into the near-optimal regions, similar to the algorithms in (Shekhar and Javidi, 2018; 2020). The first property allows us to construct tighter confidence intervals, which results in the algorithm achieving the minimax regret rate. As a result, applying Theorem 1 to this algorithm provides us with the best (i.e., the highest) instance-dependent lower bounds. Additionally, the second property allows the algorithm to exploit the 'easier' problem instances when the objective function satisfies the growth condition with small $b$, and results in improved regret bound in these problem instances. This is proved in Theorem 2 and Proposition 2.

### 4.2. Regret Bound for Algorithm 1 ($\mathcal{A}_1$)

In this section, we analyze the performance of Algorithm 1 on individual problem instances lying in the RKHS associated with Matérn kernels. To do so, we introduce the following complexity measure associated with a problem instance $f$.

**Definition 8** (Upper-Complexity)**.** Consider a function $f \in \mathcal{H}_{K_\nu}$ with $\nu > 0$, and define $\xi = \min\{1, \nu\}$. For given constants $\Delta, \rho \in (0, 1)$ and $c_1, c_2 > 0$, introduce the set $\widetilde{\mathcal{Z}}_k \coloneqq \{x \in \mathcal{X} : f(x^*) - f(x) \leq c_1 (1/\rho)^{k\xi} \Delta\}$ for $k \geq 0$, and let $\widetilde{m}_k$ denote the $2\widetilde{w}_k \coloneqq 2c_2 \left( (1/\rho)^{k\xi} \Delta \right)^{1/\xi}$ packing number of the set $\widetilde{\mathcal{Z}}_k$. Define the upper-complexity term as follows:

$$\overline{\mathcal{C}}_f(\Delta) \coloneqq \sum_{k \geq 0} \frac{\widetilde{m}_k}{(1/\rho)^{k\xi} \Delta}.$$

Note that the notation $\overline{\mathcal{C}}_f(\Delta)$ suppresses the $\nu$, $\rho$, $c_1$ and $c_2$ dependence of the complexity term.

*Remark* 6 (Comparison of $\underline{\mathcal{C}}_f$ and $\overline{\mathcal{C}}_f$). The upper-complexity term introduced above has a similar form as the corresponding lower-complexity term ($\underline{\mathcal{C}}_f$), introduced earlier in Definition 3: both complexity measures sum over terms involving a packing number of a near-optimal set in the numerator, and an exponentially growing term times $\Delta$ in the denominator. Despite this similarity, the upper-complexity term is, in general, larger than the corresponding lower-complexity term. This is because $\widetilde{w}_k$ is proportional to $\Delta^{1/\xi}$, while $w_k$ (in Definition 3) is proportional to $\Delta^{1/\nu}$. Since $\nu \geq \xi \coloneqq \min\{1, \nu\}$, and $\Delta < 1$, the term $\widetilde{m}_k$ represents a much tighter packing than the corresponding term,

$m_k$ in Definition 3. Another, less important, factor in $\overline{\mathcal{C}}_f$ being larger than $\underline{\mathcal{C}}_f$ is that the set $\widetilde{\mathcal{Z}}_k$ is usually larger than the corresponding 'annular' set $\mathcal{Z}_k$ used in defining $\underline{\mathcal{C}}_f$.

We now state the main result of this section stating that the algorithm $\mathcal{A}_1$ is minimax near-optimal, and its instance-dependent regret can be characterized by the upper-complexity term defined above.

**Theorem 2.** *For the class of functions $f \in \mathcal{H}_{K_\nu}(M)$, the algorithm $\mathcal{A}_1$ is $a_0$-consistent with $a_0 = a_\nu^* = (\nu+d)/(\nu+2d)$. Furthermore, $\mathcal{A}_1$ also satisfies the following instance-dependent upper bound on the expected regret:*

$$\mathbb{E}\left[\mathcal{R}_n\left(\mathcal{A}_1, f\right)\right] = \widetilde{\mathcal{O}}\left(\overline{\mathcal{C}}_f\left(\Delta_n\right)\right),$$

*where $\Delta_n = \min\{n^{-(1-a_\nu^*)}, \rho^{H_n\xi}\}$, where $H_n$ is defined precisely in Lemma 7 in Appendix D. The $\widetilde{\mathcal{O}}(\cdot)$ term suppresses polylogarithmic factors in $n$.*

Again, it is instructive to specialize the above result to the special case of an instance $f$ that also satisfy the additional local-growth condition around its optima.

**Proposition 2.** *Consider a function $f$ satisfying Assumption 1 with $K = K_\nu$, and Assumption 2b with parameter $\sigma^2$, that also satisfies the growth condition (introduced in Definition 4) with an exponent $b$. Then, the cumulative regret of Algorithm 1 satisfies the following, with $\xi = \min\{1, \nu\}$:*

$$\mathbb{E}\left[\mathcal{R}_n\left(\mathcal{A}_1, f\right)\right] = \widetilde{\mathcal{O}}(n^a), \quad \text{where}$$
$$a \coloneqq \min\left(\frac{d+\nu}{d+2\nu}, \frac{d(1-\xi/b)^+ + \xi}{d(1-\xi/b)^+ + 2\xi}\right).$$

*The notation $(z)^+$ refers to $\max\{0, z\}$ and the notation $\tilde{\mathcal{O}}$ hides the polylogarithmic factors in the upper bound.*

The proof of this result is given in Appendix D. In particular, this result implies that for an instance $f$ with local growth exponent $b > 0$, for all values of $d \geq 1$ and $0 < \nu < \frac{1}{1-\frac{1}{b}}$, the regret achieved by Algorithm 1 is tighter than the minimax rate (achieved by `SupKernelUCB`). Furthermore, for a fixed $\nu > 0$, the amount of possible improvement increases with decreasing values of $b$.

While the regret achieved by $\mathcal{A}_1$ is better than those of existing algorithms, it is still larger than the corresponding instance-dependent lower bound as shown in Figure 1. The key reason is that $\mathcal{A}_1$ uses the Hölder continuity (with exponent $\xi$) of the elements of $\mathcal{H}_{K_\nu}$ to adaptively discretize the domain, which leads to the $\xi$ (instead of $\nu$) dependence of the complexity term $\overline{\mathcal{C}}_f$. This looseness can be avoided if we can construct tighter $\mathcal{X}$-uniform confidence intervals (Vakili et al., 2021b) for $f$.

## 5. Conclusion

In this paper, we initiated the instance-dependent analysis of the kernelized bandits problem. This investigation was motivated by the practical question of identifying the limits of performance achievable by commonly used algorithms, such as `GP-UCB`, on typical, non-adversarial, problem instances. We first obtained a general complexity measure that characterizes the fundamental hardness of a specific problem instance. Then, we specialized this result to problem-instances satisfying an additional local growth condition, to obtain explicit lower bounds in terms of the budget $n$. Finally, we introduced a new algorithm that achieves the best of both worlds: it matches the worst case performance limit (modulo polylogarithmic terms) established by prior work, but also has the ability to adapt to the easier problem instances.

The results of this paper lead to several interesting questions for future work, and we describe two key directions below:

- A natural next step is to investigate whether we can design an algorithm that is both minimax near-optimal and instance-optimal for the Matérn family. More specifically, with $a_0 = a_\nu^* \coloneqq (d+\nu)/(d+2\nu)$, can we design an algorithm that satisfies $\sup_{f\in\mathcal{H}_{K_\nu}(M)} \mathbb{E}[\mathcal{R}_n(\mathcal{A}, f)] = \widetilde{\mathcal{O}}(n^{a_0})$, and $\mathbb{E}\left[\mathcal{R}_n(\mathcal{A}, f)\right] = \widetilde{\mathcal{O}}\left(\underline{\mathcal{C}}_f(\Delta_n)\right)$ with $\Delta_n = n^{-(1-a)}$ for any $a > a_0$ simultaneously (recall that $\underline{\mathcal{C}}_f$ denotes the complexity term introduced in Definition 3). From a technical point of view, achieving this will be significantly aided by deriving tight time and $\mathcal{X}$-uniform confidence intervals for the GP model.
- Another interesting line of work is to adapt the ideas used in this paper to problems such as kernel level-set estimation, and optimization in other function spaces (Singh, 2021; Liu et al., 2021). The lower-bound technique of our paper can be easily generalized to these cases, as we discuss briefly in **??**. However, designing algorithms that match the so-obtained instance-dependent lower bounds may require new techniques.
- Finally, the focus in this paper was on obtaining instance-dependent bounds for the cumulative regret $\mathcal{R}_n$. It is interesting to explore whether similar results can be obtained for simple regret in the pure exploration setting.

## References

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

S. Bubeck, R. Munos, G. Stoltz, and C. Szepesvári. X-

armed bandits. *Journal of Machine Learning Research*, 12(5), 2011.

X. Cai and J. Scarlett. On lower bounds for standard and robust gaussian process bandit optimization. In *International Conference on Machine Learning*, pages 1216–1226. PMLR, 2021.

R. Camilleri, K. Jamieson, and J. Katz-Samuels. High-dimensional experimental design and kernel bandits. In *International Conference on Machine Learning*, pages 1227–1237. PMLR, 2021.

R. M. Castro and R. D. Nowak. Minimax bounds for active learning. *IEEE Transactions on Information Theory*, 54 (5):2339–2353, 2008.

S. R. Chowdhury and A. Gopalan. On kernelized multi-armed bandits. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 844–853. JMLR. org, 2017.

R. Degenne, P. Ménard, X. Shang, and M. Valko. Gamification of pure exploration for linear bandits. In *International Conference on Machine Learning*, pages 2432–2442. PMLR, 2020.

A. Garivier, P. Ménard, and G. Stoltz. Explore first, exploit next: The true shape of regret in bandit problems. *Mathematics of Operations Research*, 44(2):377–399, 2019.

D. Janz, D. Burt, and J. González. Bandit optimisation of functions in the matérn kernel rkhs. In *International Conference on Artificial Intelligence and Statistics*, pages 2486–2495. PMLR, 2020.

M. Kanagawa, P. Hennig, D. Sejdinovic, and B. K. Sriperumbudur. Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv preprint arXiv:1807.02582*, 2018.

R. Kleinberg, A. Slivkins, and E. Upfal. Bandits and experts in metric spaces. *Journal of the ACM (JACM)*, 66(4): 1–77, 2019.

T. Lattimore and C. Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Z. Li and J. Scarlett. Gaussian process bandit optimization with few batches. In *International Conference on Artificial Intelligence and Statistics*, pages 92–107. PMLR, 2022.

Y. Liu, Y. Wang, and A. Singh. Smooth bandit optimization: generalization to holder space. In *International Conference on Artificial Intelligence and Statistics*, pages 2206–2214. PMLR, 2021.

R. Munos. Optimistic optimization of a deterministic function without the knowledge of its smoothness. In *Advances in neural information processing systems*, pages 783–791, 2011.

Y. Polyanskiy and Y. Wu. Lecture notes on information theory. Technical report, MIT, 2014. (http://people.lids.mit.edu/yp/homepage/data/itlectures_v5.pdf).

A. Ramdas and A. Singh. Optimal rates for stochastic convex optimization under tsybakov noise condition. In *Proceedings of the 30th International Conference on Machine Learning*, pages 365–373. PMLR, 2013.

S. Salgia, S. Vakili, and Q. Zhao. A domain-shrinking based bayesian optimization algorithm with order-optimal regret performance. *Advances in Neural Information Processing Systems*, 34, 2021.

J. Scarlett, I. Bogunovic, and V. Cevher. Lower bounds on regret for noisy gaussian process bandit optimization. In *Conference on Learning Theory*, pages 1723–1742, 2017.

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

S. Shekhar and T. Javidi. Multi-scale zero-order optimization of smooth functions in an rkhs. *arXiv preprint arXiv:2005.04832*, 2020.

S. Singh. Continuum-armed bandits: a function space perspective. In *International Conference on Artificial Intelligence and Statistics*, pages 2620–2628. PMLR, 2021.

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

M. L. Stein. *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media, 2012.

A. Tirinzoni, M. Pirotta, M. Restelli, and A. Lazaric. An asymptotically optimal primal-dual incremental algorithm for contextual linear bandits. *Advances in Neural Information Processing Systems*, 33:1417–1427, 2020.

S. Vakili, K. Khezeli, and V. Picheny. On information gain and regret bounds in gaussian process bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 82–90. PMLR, 2021a.

S. Vakili, J. Scarlett, and T. Javidi. Open problem: Tight online confidence intervals for rkhs elements. In *Conference on Learning Theory*, pages 4647–4652. PMLR, 2021b.

M. Valko, N. Korda, R. Munos, I. Flaounas, and N. Cristianini. Finite-time analysis of kernelised contextual bandits. In *Uncertainty in Artificial Intelligence*, page 654, 2013.

R. van Handel. Probability in high dimension. Technical report, Princeton University NJ, 2014. (https://web.math.princeton.edu/~rvan/APC550.pdf).

Z. Wang, B. Shakibi, L. Jin, and N. Freitas. Bayesian multi-scale optimistic optimization. In *Artificial Intelligence and Statistics*, pages 1005–1014. PMLR, 2014.

## A. Additional Definitions

In this section, we list the definitions of several technical terms that have been used in stating the main results of the paper. We begin with the definition of a positive definite kernel, that will then be used in defining an RKHS.

**Definition 9** (Positive Definite Kernel)**.** For a non-empty set $\mathcal{X}$, a symmetric function $K : \mathcal{X} \times \mathcal{X} \mapsto [0, \infty)$ is called a positive-definite kernel, if for any $m \in \mathbb{N}$, any $x_1, \ldots, x_m \in \mathcal{X}$ and $c_1, c_2, \ldots, c_m \in \mathbb{R}$, the following is true: $\sum_{i=1}^{m} \sum_{j=1}^{m} c_i K(x_i, x_j) c_j \geq 0$.

In this paper, we will focus primarily on a family of kernels, referred to the Matérn family, that are parameterized by a smoothness parameter $\nu > 0$.

**Definition 10** (Matérn kernels)**.** For $\nu > 0$ and $\theta > 0$, the Matérn kernel $K_\nu : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is defined as

$$K_\nu(x, z) = \frac{1}{2^{\nu-1}\Gamma(\nu)} \left( \frac{\sqrt{2\nu}\|x - z\|}{\theta} \right)^{\nu} J_\nu \left( \frac{\sqrt{2\nu}\|x - z\|}{\theta} \right),$$

where $J_\nu$ denotes the modified Bessel function of the second kind of order $\nu$.

The RKHS associated with Matérn kernels consist of functions with a 'finite degree of smoothness' (Kanagawa et al., 2018) as opposed to the infinitely differentiable functions lying the RKHS associated with the SE kernels. Due to this property, the Matérn kernels are commonly used in practical problems (Stein, 2012, § 1.7) as they provide a reasonable trade-off between analytical tractability and representation power.

We now present a formal definition of the RKHS associated with a positive definite kernel $K$.

**Definition 11** (RKHS)**.** For a nonempty set $\mathcal{X}$ and a positive-definite kernel $K$, the RKHS associated with $K$, denoted by $\mathcal{H}_K$, is defined as the Hilbert space of functions on $\mathcal{X}$ with an inner product $\langle \cdot, \cdot \rangle$ satisfying the following: **(i)** for all $x \in \mathcal{X}$, the function $K(\cdot, x) \in \mathcal{H}_K$, and **(ii)** for all $x \in \mathcal{X}$ and $g \in \mathcal{H}_K$, we have $g(x) = \langle g, K(\cdot, x) \rangle$.

The equality $g(x) = \langle g, K(\cdot, x) \rangle$ is referred to as the *reproducing property* which lends the name to the RKHS. We next introduce the definition of Gaussian Processes (GPs) that are often used as a surrogate model for estimating functions lying in an RKHS.

**Definition 12** (Gaussian Processes)**.** For a positive definite kernel $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$, we use $GP(0, K)$ to represent a stochastic process indexed by $\mathcal{X}$, denoted by $\{Z_x : x \in \mathcal{X}\}$, such that for any $m \in \mathbb{N}$ and $x_1, \ldots, x_m \in \mathcal{X}$, the random vector $[Z_{x_1}, \ldots, Z_{x_m}] \sim N(0, \Sigma_m)$ with $\Sigma_m = [K(x_i, x_j)]_{1 \leq i,j \leq m}$.

Next, we present the formal definition of the probability measure that the noisy zeroth-order-oracle and adaptive sampling scheme induce on the space $([0, 1]^d \times \mathbb{R})^n$. This term is used in Lemma 1.

**Definition 13** (Induced Probability Measure)**.** An adaptive sampling strategy $\mathcal{A}$ and a function $f$ induces a probability measure $\mathbb{P}_{f,\mathcal{A}}$ (henceforth abbreviated as $\mathbb{P}_f$) on the measurable space $(\Omega, \mathcal{F})$, with $\Omega = (\mathcal{X} \times \mathcal{Y})^n$ and $\mathcal{F}$ representing the Borel $\sigma-$algebra on $\Omega$. Recall that we have $\mathcal{X} = [0, 1]^d$ and $\mathcal{Y} = \mathbb{R}$. This measure assigns the probabilities to events $E = \prod_{t=1}^{n} (E_{t,X} \times E_{t,Y})$ where $E_{t,X} \in \mathcal{B}_\mathcal{X}$ and $E_{t,Y} \in \mathcal{B}_\mathcal{Y}$.

$$\mathbb{P}_f\big((X^n, Y^n) \in E\big) = \prod_{t=1}^{n} P_f\left(Y_t \in E_{t,Y} | X^t, Y^{t-1}\right) P_\mathcal{A}\left(X_t \in E_{t,X} | X^{t-1}, Y^{t-1}\right).$$

Here $P_f$ represents the noisy zeroth-order-oracle and $P_\mathcal{A}$ is determined by the sampling strategy.

We end this section with the definitions of sub-Gaussian random variables (used in stating Assumption 2b) and packing numbers (used in defining the complexity terms $\underline{\mathcal{C}}_f$ and $\overline{\mathcal{C}}_f$).

**Definition 14** (Sub-Gaussianity)**.** A random variable $X$ is said to be sub-Gaussian with parameter $\sigma^2$ if it satisfies the following for all $t \in \mathbb{R}$:

$$\mathbb{E}[e^{tX}] \leq \exp\left(\frac{\sigma^2 t^2}{2}\right).$$

**Definition 15** (Packing number)**.** Given a subset $S$ of a metric space $(\mathcal{X}, \ell)$, the $w$ packing number of $S$ is the cardinality of the largest $E \subset S$ such that any two $z, z' \in E$ satisfy $\ell(e, e') \geq w$.

## B. Proof of Theorem 1

We present a detailed outline of the proof of Theorem 1 in Appendix B.1, and describe the formal steps of the proof in Appendix B.2 and Appendix B.3. In particular, Appendix B.2 has the statement and proof of an intermediate result that gives us a single term in the lower bound of Theorem 1, and Appendix B.3 contains the steps required to obtain the statement of Theorem 1 from the intermediate result.

### B.1. Overview of the argument

We begin by presenting an informal description of the key ideas involved in the obtaining the main lower bound, formally stated as Theorem 1 in Section 3.1.

Suppose $f$ is a function lying in $\mathcal{H}_{K_\nu}(M)$ and let $\mathcal{A}$ be an $a_0$-consistent algorithm for this family of functions. Suppose $E_1, E_2, \ldots, E_m$ are $m$ disjoint subsets of the input space $\mathcal{X}$ for some $m \geq 1$, with the property that

$$f(x) \leq f(x^*) - \Delta_i, \quad \text{for all } x \in E_i, \quad \text{for all } i \in [m]. \tag{4}$$

Now if $N_i$ denotes the (random) number of times the algorithm $\mathcal{A}$ queries points in the region $E_i$ in $n$ rounds, then we immediately have the following regret lower bound.

$$\mathbb{E}_f[\mathcal{R}_n(\mathcal{A}, f)] \geq \sum_{i=1}^{m} \Delta_i \mathbb{E}_f[N_i]. \tag{5}$$

The expression in (5) suggests that one way of lower bounding the regret incurred by $\mathcal{A}$ on the function $f$ is to lower bound the expected number of samples it allocates in these suboptimal regions, $\mathbb{E}_f[N_i]$ for $1 \leq i \leq m$. We approach this task by considering functions that are slightly perturbed versions of $f$, denoted by $\tilde{f}_i$, that also lie in the same RKHS. The perturbed function $\tilde{f}_i$ shall differ from $f$ only in the region $E_i$, but this difference should be substantial enough to ensure that the maximizer of $\tilde{f}_i$ lies in $E_i$. This fact makes $\tilde{f}_i$ operationally distinct from $f$, for which the region $E_i$ is at least $\Delta_i$-suboptimal as assumed in (4). Now, since the algorithm $\mathcal{A}$ is $a_0$-consistent for the given function class, it must achieve $o(n^a)$ regret (for any $a > a_0$) for all such functions (and in particular, for $f$ and all of $\tilde{f}_i$). These two facts will enable us to bound the number of samples that the algorithm $\mathcal{A}$ must spend (on an average) in the suboptimal region $E_i$ when $f$ is the true function.

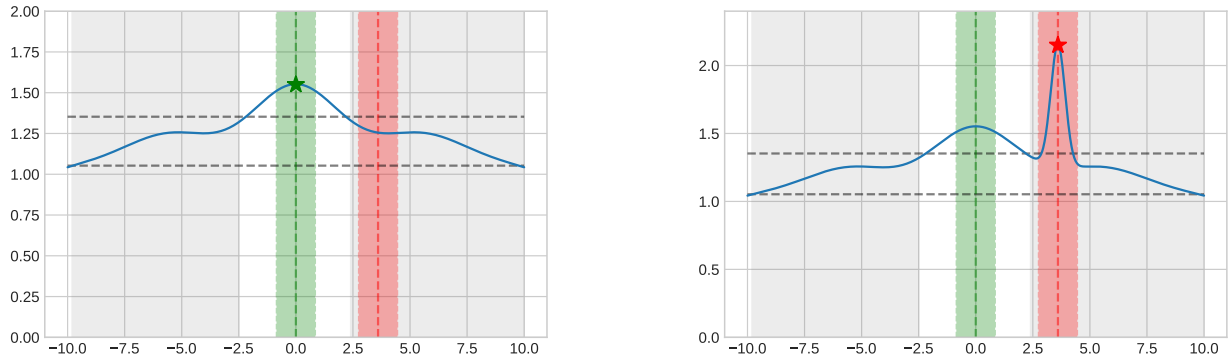


*Figure 2.* The two figures show a function $f$ (left) and its perturbed version $\tilde{f}$) (right). In the left figure, the optimum of $f$ is denoted by the green $\star$. The red shaded region corresponds to the set $E$ of Definition 16 that is disjoint from a neighborhood (green shaded region) of the optimum of $f$. The gray shaded regions represent the set $\widetilde{\mathcal{X}}(f, \Delta, c) = \{x : \Delta \leq f(x) - f(x^*) \leq c\Delta\}$ for $\Delta = 0.2$ and $c = 2.5$. The figure on the right, $\tilde{f}$, is then obtained by adding a bump function supported on $E$ to $f$ By construction, in the left figure, the red-shaded region is at least $\Delta-$suboptimal, while in the right figure the green-shaded region is at-least $\Delta-$suboptimal. The two functions $f$, and $\tilde{f}$ will induce statistically similar distributions as they differ only in a small region (shown in red). Hence, if $f$ is the true objective function, any good algorithm $\mathcal{A}$ must spend some queries in the red region to discard the possibility that the objective function is $\tilde{f}$.

The rest of this section is organized as follows:

- In Appendix B.1.1, we describe the details of the argument in deriving a lower bound on the expected number of samples $\mathcal{A}$ spends in a suboptimal region for one perturbed function.
- In Appendix B.1.2, we present the details involved in constructing an appropriate collection of perturbed functions. In particular, this involves carefully balancing several trade-offs in the choice of parameters such as $\Delta_i$ and $E_i$, such that the resulting lower bound is tightest.

### B.1.1. A PERTURBATION ARGUMENT

Suppose $f \in \mathcal{H}_{K_\nu}(M)$ and $\mathcal{A}$ is an $a_0$-consistent algorithm. Let $\tilde{f}$ be another function in $\mathcal{H}_{K_\nu}(M)$ that is an $(E, c, \Delta)-$perturbation of $f$, as defined below.

**Definition 16** ($(E, c, \Delta)-$perturbation)**.** We say that a function $\tilde{f} \in \mathcal{H}_{K_\nu}(M)$ is an $(E, c, \Delta)$-perturbation of another function $f \in \mathcal{H}_{K_\nu}(M)$, if it satisfies the following properties:

**(P1)** $\tilde{f}$ differs from $f$ only in a region $E$ of the input space, i.e., $\tilde{f}(x) = f(x)$ for all $x \in \mathcal{X} \setminus E$.

**(P2)** The function $f$ achieves its maximum value (denoted by $f^*$) at a point $x^* \in \mathcal{X} \setminus E$. On the other hand, $\tilde{f}$ achieves its maximum value $\tilde{f}^*$ at a point $\tilde{x}^*$ lying in the region $E$.

**(P3)** There exist constants $c > 1$ and $\Delta > 0$ such that the following conditions are satisfied:

$$\begin{aligned} &|f(x) - \tilde{f}(x)| \leq c\Delta, \quad \text{for all } x \in E, \\ &f^* - f(x) \geq \Delta, \quad \text{for all } x \in E. \\ &\tilde{f}^* - \tilde{f}(x) \geq \Delta, \quad \text{for all } x \in \mathcal{X} \setminus E. \end{aligned} \tag{6}$$

*Remark* 7. In this section, we do not address the issue of existence of a function $\tilde{f}$ satisfying all the properties, as well as the possible values of $c$, $\Delta$ and choices of the region $E$. We present the argument under the assumption that such an $\tilde{f}$ exits for some fixed $c$, $\Delta$ and $E$. The trade-offs involved in constructing such $\tilde{f}$ will be discussed in Appendix B.1.2.

Similar to the more general case of (5), the definition above motivates a simple decomposition of the regret in terms of the number of queries made by $\mathcal{A}$ in the region $E$ in which $f$ and $\tilde{f}$ differ. In particular, if $N$ denotes the (random) number of times $\mathcal{A}$ queries points in $E$, then we immediately have the following:

$$\mathbb{E}_f\left[\mathcal{R}_n\left(\mathcal{A}, f\right)\right] \geq \mathbb{E}_f\left[N\right]\Delta, \qquad \textit{and} \qquad \mathbb{E}_{\tilde{f}}\left[\mathcal{R}_n(\mathcal{A}, \tilde{f})\right] \geq \mathbb{E}_{\tilde{f}}\left[n - N\right]\Delta.$$

To obtain a lower bound on $\mathbb{E}_f[N]$, we will use the following two key properties of the pair $(f, \tilde{f})$ as encoded by the formal statements in Definition 16.

- From a statistical point of view, the two problem instances are close. In particular, $f$ and $\tilde{f}$ only differ over the region $E$, and furthermore their deviation is upper bounded by $c\Delta$. This allows us to upper bound the KL divergence between their induced probability distributions $\mathbb{P}_f$ and $\mathbb{P}_{\tilde{f}}$ (see Definition 13) in terms of $\mathbb{E}_f[N]$.
- In an operational sense, the two problem instances $\tilde{f}$ and $f$ are sufficiently distinct. This is a consequence of properties (P2) and (P3) in Definition 16, which say that the optimizer of $f$ (resp. $\tilde{f}$) lies in the region $\mathcal{X} \setminus E$ (resp. E) that is known to be at least $\Delta-$suboptimal for $\tilde{f}$ (resp. $f$). This, along with the $a_0$-consistency of $\mathcal{A}$ will be used to lower-bound the KL-divergence between $\mathbb{P}_f$ and $\mathbb{P}_{\tilde{f}}$ by a constant.

Combining the two inequalities will give us the required lower bound on $\mathbb{E}_f[N]$, and consequently on $\mathbb{E}_f[\mathcal{R}_n\left(\mathcal{A}, f\right)]$. We now describe the steps.

**Step 1: Upper bound on $D_{KL}(\mathbb{P}_f, \mathbb{P}_{\tilde{f}})$.** Recall that the pairs $(f, \mathcal{A})$ and $(\tilde{f}, \mathcal{A})$ both induce a probability measure on the $n-$fold product of input-observation space $\Omega := (\mathcal{X} \times \mathbb{R})^n$. Denote the two probability measures by $\mathbb{P}_f$ and $\mathbb{P}_{\tilde{f}}$. Then, assuming that the observation noise is i.i.d. $N(0, \sigma^2)$, it can be show that

$$D_{KL}\left(\mathbb{P}_f, \mathbb{P}_{\tilde{f}}\right) \leq \frac{1}{2\sigma^2}\left(\sup_{x \in E} \tilde{f}(x) - f(x)\right)^2 \mathbb{E}_f\left[N\right] \leq \frac{c^2\Delta^2}{2\sigma^2}\mathbb{E}_f\left[N\right]. \tag{7}$$

To obtain (7), we use the closed-form expression of the KL-divergence between two univariate Gaussian random variables with the same variance, and the fact that the maximum deviation between $f$ and $\tilde{f}$ is upper bounded by $c\Delta$. The details are presented while proving Lemma 1 in Appendix B.2.1.

The statement of (7) implies that the KL-divergence between $\mathbb{P}_f$ and $\mathbb{P}_{\tilde{f}}$ can be controlled by two terms: **(i)** the maximum deviation between $\tilde{f}$ and $f$, a quantity that is bounded by $c\Delta$ by assumption, and **(ii)** the expected number of queries made by $\mathcal{A}$ in the region $E$ for the function $f$. This quantity cannot be too large either, as the algorithm $\mathcal{A}$ is assumed to be $a_0-$consistent, and the region $E$ is at least $\Delta-$suboptimal for $f$ as stated in (6).

**Step 2: Lower Bound on $D_{KL}(\mathbb{P}_f, \mathbb{P}_{\tilde{f}})$.** Since the algorithm $\mathcal{A}$ is assumed to be $a_0-$consistent, we immediately have the following two statements for any $a > a_0$:

$$\mathbb{E}_f\left[N\Delta\right] \leq \mathbb{E}_f\left[\mathcal{R}_n\left(f, \mathcal{A}\right)\right] = o\left(n^a\right), \quad \text{and} \quad \mathbb{E}_{\tilde{f}}\left[\left(n-N\right)\Delta\right] \leq \mathbb{E}_{\tilde{f}}\left[\mathcal{R}_n\left(\tilde{f}, \mathcal{A}\right)\right] = o\left(n^a\right).$$

Together, these two conditions imply that $\mathbb{E}_f[N]$ and $\mathbb{E}_{\tilde{f}}[n-N]$ cannot be too large. To see this, define a $[0,1]$ valued random variable $Z = N/n$ and let $p = \mathbb{E}_f[Z]$ and $q = \mathbb{E}_{\tilde{f}}[Z]$. Note that $Z$ denotes the fraction of samples spent in the region $E$ by the algorithm $\mathcal{A}$. Then, for large enough values of $n$, and $\Delta$ fixed, we expect that $p \approx 0$ and $q \approx 1$. This in turn implies that the KL-divergence between two Bernoulli random variables with expected values $p$ and $q$ respectively is non-zero. More specifically, we can show that there exists a constant $C > 0$ such that

$$0 < C \leq d_{KL}(p,q) = d_{KL}\left(\mathbb{E}_f\left[\frac{N}{n}\right], \mathbb{E}_{\tilde{f}}\left[\frac{N}{n}\right]\right), \tag{8}$$

where $d_{KL}(p,q) = p\log(p/q) + (1-p)\log\left((1-p)/(1-q)\right)$ denotes the KL-divergence between two Bernoulli random variables. The next step in obtaining a lower bound on $\mathbb{E}_f[N]$ is the observation that

$$d_{KL}\left(\mathbb{E}_f\left[\frac{N}{n}\right], \mathbb{E}_{\tilde{f}}\left[\frac{N}{n}\right]\right) \leq D_{KL}\left(\mathbb{P}_f, \mathbb{P}_{\tilde{f}}\right). \tag{9}$$

This is a consequence of data-processing inequality, as shown by Garivier et al. (2019).

**Step 3: Lower bound $\mathbb{E}_f[N]$.** Finally, we can use (9) to link the statements of (7) and (8), and obtain the inequality

$$C < \frac{c^2\Delta^2}{2\sigma^2}\mathbb{E}_f\left[N\right], \quad \text{which implies} \quad \mathbb{E}_f\left[N\right] \geq \frac{2C\sigma^2}{c^2\Delta^2}.$$

Multiplying this term with $\Delta$ gives us one term in the regret decomposition of (5).

### B.1.2. Constructing an $(E, c, \Delta)$ perturbed function

Next, we introduce the definition of a bump function used in (Cai and Scarlett, 2021, Lemma 4) that will be used to construct the local perturbations (satisfying the conditions of Definition 16) in our lower bound proof.

**Definition 17** (bump function $g$)**.** Define the function $g(x) = \exp\left(1 - \frac{1}{1-\|x\|^2}\right)\mathbb{1}_{\{\|x\|<1\}}$, which satisfies the following properties:

- $g$ is supported on the ball $B(0,1)$.
- $\sup_{x\in B(0,1)}\ g(x) = g(0) = 1$.
- $\|g\|_{\mathcal{H}_{K_\nu}} := M_\nu < \infty$ for some constant $M_\nu$ depending on $\nu$.
- if $\tilde{g}(\cdot) = g(\frac{\cdot}{w})$ for some $w > 0$, then $\|\tilde{g}\|_{\mathcal{H}_{K_\nu}} \leq (1/w)^\nu M_\nu$.

We now discuss the details of constructing a function $\tilde{f}$ satisfying the conditions of Definition 16. Here is the summary for a fixed $a \in (a_0, 1)$.

- The term $\Delta$ should not be smaller than $n^{-(1-a)}$.

- An appropriate choice of the set $E$ is a ball $B(z, w)$ for a point $z \in \mathcal{X} \setminus \mathcal{X}^*$ and radius $w > 0$. Recall that $\mathcal{X}^* = \{x^* \in \mathcal{X} : f(x^*) = f^* := \max_{x \in \mathcal{X}} f(x)\}$.
- We define the perturbed function $\tilde{f} = f + \tilde{g}$ where $\tilde{g}(\cdot) = (c+1)g\left(\frac{\cdot - z}{w}\right)$ for some $w > 0$. Here $g$ is the bump function introduced in Definition 17. The specific constraints on $w$ are discussed below.

We now discuss these choices in more details.

**Choice of $\Delta$.** The term $\Delta$ parametrizes the amount of perturbation between $f$ and $\tilde{f}$. $\Delta$ should be large enough to ensure that $\tilde{f}$ and $f$ are *distinguishable* from the point of view of the algorithm $\mathcal{A}$. In particular, fix any $a > a_0$. Then for this value of $a$, $\Delta$ must be larger than $n^{-(1-a)}$. This is because, if $\Delta < n^{-(1-a)}$, then the algorithm may spend all $n$ of its samples in the region $E$ under $f$ as well as $\tilde{f}$ without violating the $o(n^a)$ requirement on regret.

**Choice of $z$ and $w$.** The terms $z$ and $w$ must be such that $B(z, w) \subset \widetilde{\mathcal{X}}(f, \Delta, c) := \{x \in \mathcal{X} : \Delta \leq f^* - f(x) \leq c\Delta\}$. Thus $z$ and $w$ must be selected to ensure that the ball of radius $w$ around $z$ is fully contained in the *annular* region $\widetilde{\mathcal{X}}(f, \Delta, c)$ in which the sub-optimality of $f$, (i.e., $f^* - f(x)$) is between $\Delta$ and $c\Delta$. Furthermore, the radius $w$ must be large enough to satisfy the condition stated in (10) below.

**Defining $\tilde{f}$.** Having defined the region $E$, we then construct the perturbed version of $f$, by adding a shifted and scaled bump function to it. In particular, we add $\tilde{g} = (c+1)\Delta g\left(\frac{\cdot - z}{w}\right)$ to $f$. Note that, by assumption, the RKHS norm of $f$ satisfies $\|f\|_{\mathcal{H}_{K_\nu}} < M$. Since we require the perturbed function to also lie in the class $\mathcal{H}_{K_\nu}(M)$, a sufficient condition for that is

$$\|\tilde{g}\|_{\mathcal{H}_{K_\nu}} \leq \frac{(c+1)\Delta}{w^\nu}\|g\|_{\mathcal{H}_{K_\nu}} = \frac{(c+1)\Delta}{w^\nu}M_\nu \leq M - \|f\|_{\mathcal{H}_{K_\nu}} \quad \Rightarrow \; w \geq \left(\frac{(c+1)\Delta M_\nu}{M - \|f\|_{\mathcal{H}_{K_\nu}}}\right)^{1/\nu}. \tag{10}$$

*Remark* 8. Note that the expression for $w$ in (10) implicitly assumes that for this value of $c$ and $\Delta$ the region $\widetilde{\mathcal{X}}(f, \Delta, c)$ is large enough to contain a ball of this (or larger) radius. Our result, Theorem 3, holds under this assumption. In case this condition is violated, Theorem 3 reduces to the trivial lower bound $\mathbb{E}_f[\mathcal{R}_n(f, \mathcal{A})] \geq 0$.

### B.2. An intermediate one-step result

**Proposition 3.** *Consider the kernelized bandit problem with a budget $n$ and objective function $f \in \mathcal{H}_{K_\nu}(M)$ with $\|f\|_{\mathcal{H}_{K_\nu}} = (1-\lambda)M$ for some $\lambda \in (0,1)$. Let $\mathcal{A}$ denote an $a_0$-consistent algorithm for the class $\mathcal{H}_{K_\nu}(M)$, and fix an $a > a_0$. For constants $\Delta \geq 16n^{-(1-a)}$ and $c > 1$, introduce the set $\mathcal{Z} = \{x \in \mathcal{X} : \Delta \leq f(x^*) - f(x) < c\Delta\}$, and with $w = ((c+1)\Delta M_\nu/(M\lambda))^{1/\nu}$, use $m(\mathcal{Z}, w)$ to denote the $2w$ packing number of $\mathcal{Z}$. Then, the following is true for $n$ large enough (exact condition in equation 13 below):*

$$\mathbb{E}\left[\mathcal{R}_n(\mathcal{A}, f)\right] \geq \frac{7\log 2}{4}\,\frac{m(\mathcal{Z}, w)\,\sigma^2}{c^2\Delta}. \tag{11}$$

*Proof.* Let $\{z_i : 1 \leq i \leq m(\mathcal{Z}, w)\}$ denote the points that form the maximal $2w$ packing set of $\mathcal{Z}$ with cardinality $m = m(\mathcal{Z}, w)$. By definition of $\mathcal{Z}$, the region $B(z_i, w)$ is at least $\Delta$-suboptimal for $f$. Building upon this fact, the proof of the result follows in these three steps:

- First, we show that we can construct $m$ perturbed functions, denoted by $\{f_i : 1 \leq i \leq m(\mathcal{Z}, w)\}$, as introduced in Definition 16. The function $f_i$ differs from $f$ only in the region $B(z_i, w)$, and in fact, it achieves its maximum value in that region.
- Next, for each such perturbed function, we obtain a lower bound on the number of samples that the algorithm $\mathcal{A}$ must spend in $B(z_i, w)$.
- Finally, the result follows by using the regret decomposition Equation (5), again using the fact that the points in $B(z, w)$ are $\Delta$-suboptimal for $f$.

We now present the details of the steps outlined above. For every $i \in \{1, \ldots, m\}$, define the function $f_i = f + g_i$, where $g_i = (c+1)\Delta g\left(\frac{\cdot - z_i}{w}\right)$ is the scaled and shifted version of the bump function introduced in Definition 17. Now the choice

of the radius (or scale parameter) $w$ in the definition of $f_i$ according to (10) implies that

$$\|g_i\|_{\mathcal{H}_{K_\nu}} \leq \frac{(c+1)\Delta}{w^\nu} M_\nu \leq \lambda M.$$

This implies the following:

- The function $f_i$ satisfies $\|f_i\|_{\mathcal{H}_{K_\nu}} \leq \|f\|_{\mathcal{H}_{K_\nu}} + \|g_i\|_{\mathcal{H}_{K_\nu}} \leq M$. Thus, the function $f_i$ lies in the class $\mathcal{H}_{K_\nu}(M)$, and hence $\mathcal{A}$ achieves a regret $o\left(n^a\right)$ for any $a > a_0$ on $f_i$ for all $1 \leq i \leq m$.
- The functions $f$ and $f_i$ have well separated optimal regions. More formally, if $x^*$ and $x_i^*$ denote the maximizers of $f$ and $f_i$ respectively, the following are true:

$$f(x^*) - f(x) \geq \Delta, \quad \text{for all } x \in B(z_i, w),$$
$$f_i(x_i^*) - f_i(x) \geq \Delta, \quad \text{for all } x \notin B(z_i, w).$$

- The functions $f$ and $f_i$ differ from each other only in the region $B(z_i, w)$ and furthermore, they satisfy the following uniform deviation bound:

$$\sup_{x \in \mathcal{X}} |f(x) - f_i(x)| \leq c\Delta.$$

To summarize the above three points, the function $f_i$ is a $\big(B(z_i, w), \Delta, c\big)$-perturbation of $f$. Next, we show that any $a_0$-consistent algorithm must allocate at least a certain number of points to the region $B(z_i, w)$ when the true underlying function is $f$, in order to gather enough evidence to reject $f_i$.

**Lemma 1.** *Let $N_i(\mathcal{A}, n)$ denote the number of times the algorithm $\mathcal{A}$ queries the oracle at points in the region $B(z_i, w)$. Then we have the following bound:*

$$\mathbb{E}_f\left[N_i(\mathcal{A}, n)\right] \geq \frac{2\sigma^2}{c^2\Delta^2}\left((1 - p_{n,i}) \log\left(\frac{1}{1 - q_{n,i}}\right) - \log 2\right), \quad \textit{where} \tag{12}$$
$$p_{n,i} \coloneqq \frac{\mathbb{E}_f\left[N_i(\mathcal{A}, n)\right]}{n} \quad \textit{and} \quad q_{n,i} \coloneqq \frac{\mathbb{E}_{f_i}\left[N_i(\mathcal{A}, n)\right]}{n}.$$

*In the above display, $\mathbb{E}_f$ denotes the expectation w.r.t. the probability measure induced by the pair $(f, \mathcal{A})$, and similarly $\mathbb{E}_{f_i}$ denotes the probability measure induced by the pair $(f_i, \mathcal{A})$ for $1 \leq i \leq m$.*

The proof of (12) follows by relating the regret incurred by $\mathcal{A}$ on $f$ and $f_i$ respectively to a pair of multi-armed bandit problems with $(m+1)$ arms, and then applying the *fundamental information inequality* (Garivier et al., 2019, § 2). The details of this proof are deferred to Appendix B.2.1

Next, we simplify the expression obtained in Lemma 1 by appealing to the $a_0$-consistency of the algorithm $\mathcal{A}$. In the process, we also clarify the meaning of the assumption that "$n$ is large enough" in the statement of Theorem 3. In particular, we require that $n$ is large enough to ensure the following to hold simultaneously

$$\mathbb{E}_f\left[\mathcal{R}_n\left(\mathcal{A}, f\right)\right] \leq 2n^a, \quad \text{and} \quad \mathbb{E}_{f_i}\left[\mathcal{R}_n\left(\mathcal{A}, f_i\right)\right] \leq 2n^a, \quad \text{for all } i \in [m]. \tag{13}$$

More specifically, using the notation $f_0 = f$, we note that due to the $a_0$-consistency of $\mathcal{A}$, and the fact that all the functions $\{f_i : 0 \leq i \leq m\}$ lie in $\mathcal{H}_{K_\nu}(M)$, we must have $\lim_{n\to\infty} \mathbb{E}_{f_i}[\mathcal{R}_n(\mathcal{A}, f_i)]/n^a = 0$. Hence, there must exist a finite $n_0$ such that for all $n \geq n_0$, we have $\mathbb{E}_{f_i}[\mathcal{R}_n(\mathcal{A}, f_i)] \leq 2n^a$ for all $i \in \{0, 1, \ldots, m\}$.

Next, we observe that

$$1 - p_{n,i} = 1 - \frac{\mathbb{E}_f\left[N_i(\mathcal{A}, n)\right]}{n} = 1 - \frac{\Delta\mathbb{E}_f\left[N_i(\mathcal{A}, n)\right]}{\Delta n}$$
$$\geq 1 - \frac{\mathbb{E}_f\left[\mathcal{R}_n(\mathcal{A}, f)\right]}{\Delta n} \geq 1 - \frac{\mathbb{E}_f\left[\mathcal{R}_n(\mathcal{A}, f)\right]}{16n^{1-(1-a)}} \tag{14}$$
$$\geq \frac{7}{8}. \tag{15}$$

In the above display,
(14) uses the fact that $N_i(\mathcal{A}, n)\,\Delta \leq \mathcal{R}_n(\mathcal{A}, f)$, and that $\Delta \geq 16n^{-(1-a)}$,
(15) uses the assumption made in (13) that $n$ is large enough to ensure that $\mathbb{E}_0[\mathcal{R}_n(\mathcal{A}, f)] \leq 2n^a$.

Similarly, for the $q_{n,i}$ dependent term, we have

$$\frac{1}{1-q_{n,i}} = \frac{n}{n - \mathbb{E}_{f_i}[N_i(\mathcal{A}, n)]} = \frac{n\Delta}{(n - \mathbb{E}_{f_i}[N_i(\mathcal{A}, n)])\,\Delta} \geq \frac{n\Delta}{\mathbb{E}_{f_i}[\mathcal{R}_n(\mathcal{A}, f_i)]} \geq \frac{n\Delta}{2n^a} \tag{16}$$

$$= \frac{n^{1-a}\Delta}{2}. \tag{17}$$

In the above display, (16) uses the assumption that $n$ is large enough to ensure that $\mathbb{E}_{f_i}[\mathcal{R}_n(\mathcal{A}, f_i)] \leq 2n^a$.

Putting (15) and (17) back in (12), we get

$$\mathbb{E}_f[N_i(\mathcal{A}, n)] \geq \frac{2\sigma^2}{c^2\Delta^2}\left(\frac{7}{8}\log\left(\frac{\Delta n^{1-a}}{2\,2^{8/7}}\right)\right) \geq \frac{7\log 2}{4}\frac{\sigma^2}{c^2\Delta^2}.$$

Finally, the result stated in (11) follows by repeating the argument of Lemma 1 for all the different values of $i \in \{1, \ldots, m\}$, and noting that $\mathbb{E}_f[\mathcal{R}_n(\mathcal{A}, f)] \geq \sum_{i=1}^m \Delta\mathbb{E}_f[N_i(\mathcal{A}, n)]$ from the decomposition inequality (5). □

### B.2.1. Proof of Lemma 1

To prove this result, we need to introduce some additional notation. We use $\mathcal{H}_t$ to denote the observations up to, and including, time $t$ for $t \in \{1, 2, \ldots, n\}$. For a given $n \geq 1$, introduce the sample space $\Omega = (\mathcal{X} \times \mathcal{Y})^n$, and let $\mathcal{F}_0$ denote a sigma algebra of subsets of $\Omega$. For a given function $f$ and an querying strategy $\mathcal{A}$, we use $\mathbb{P}_{f,\mathcal{A}}$ to denote the probability measure on $\Omega$ induced by the pair $(f, \mathcal{A})$. We will drop the $\mathcal{A}$ dependence of $\mathbb{P}_{f,\mathcal{A}}$, and simply use $\mathbb{P}^{(f)}$ in the sequel.

The first step is to obtain an upper bound on the KL-divergence between the measures $\mathbb{P}^{(f)}$ and $\mathbb{P}^{(f_i)}$ induced on the space $\Omega$, for a common algorithm $\mathcal{A}$. In particular, suppose $(X_1, Y_1, \ldots, X_n, Y_n)$ denote the query-observation pairs collected by the algorithm $\mathcal{A}$ up to time $n$. Then we have the following:

$$\begin{aligned}
\mathcal{D}_n &:= D_{KL}\left(\mathbb{P}^{(f)}, \mathbb{P}^{(f_i)}\right) \\
&= \mathcal{D}_{n-1} + D_{KL}\left(\mathbb{P}^{(f)}_{X_n,Y_n|\mathcal{H}_{n-1}}, \mathbb{P}^{(f_i)}_{X_n,Y_n|\mathcal{H}_{n-1}}\right) && (18)\\
&= \mathcal{D}_{n-1} + D_{KL}\left(\mathbb{P}^{(f)}_{X_n|\mathcal{H}_{n-1}}, \mathbb{P}^{(f_i)}_{X_n|\mathcal{H}_{n-1}}\right) + D_{KL}\left(\mathbb{P}^{(f)}_{Y_n|\mathcal{H}_{n-1},X_n}, \mathbb{P}^{(f_i)}_{Y_n|\mathcal{H}_{n-1},X_n}\right) && (19)\\
&= \mathcal{D}_{n-1} + 0 + D_{KL}\left(\mathbb{P}^{(f)}_{Y_n|\mathcal{H}_{n-1},X_n}, \mathbb{P}^{(f_i)}_{Y_n|\mathcal{H}_{n-1},X_n}\right) && (20)\\
&= \mathcal{D}_{n-1} + \mathbb{E}_f\left[\frac{(f(X_n) - f_i(X_n))^2}{2\sigma^2}\right] && (21)\\
&\leq \mathcal{D}_{n-1} + \mathbb{E}_f\left[\mathbb{1}_{\{X_n \in B(z_i,w)\}}\frac{c^2\Delta^2}{2\sigma^2}\right] && (22)
\end{aligned}$$

In the above display,

- (18) and (19) follow from the chain rule for KL-divergence (Polyanskiy and Wu, 2014, Theorem 2.2),
- (20) uses the fact that conditioned on $\mathcal{H}_{n-1}$, the distribution of $X_n$ is the same for both the problem instances, that is they are both selected according to the mapping $A_n : (\mathcal{X} \times \mathcal{Y})^{n-1} \mapsto \mathcal{X}$, where $\mathcal{A} = (A_t)_{t=1}^n$ is the common strategy,
- (21) uses the fact that condition on $X_n$, $Y_n$ is distributed as $N\left(f(X_n), \sigma^2\right)$ and $N\left(f_i(X_n), \sigma^2\right)$ under the two distributions $\mathbb{P}^{(f)}$ and $\mathbb{P}^{(f_i)}$ respectively, and
- (22) uses the fact that, by construction, $f$ and $f_i$ only differ in the region $B(z_i, w)$, and furthermore, in this region we have $\max_{x \in B(z_i,w)} |f(x) - f_i(x)| \leq c\Delta$

Repeating the steps involved in obtaining (22) $n-1$ times, we get the following upper bound on the term $\mathcal{D}_n$:

$$\begin{aligned}\mathcal{D}_n &\leq \frac{c^2\Delta^2}{2\sigma^2}\sum_{t=1}^{n}\mathbb{E}_f\left[\mathbb{1}_{\{X_t\in B(z_i,w)\}}\right] = \frac{c^2\Delta^2}{2\sigma^2}\mathbb{E}_f\left[\sum_{t=1}^{n}\mathbb{1}_{\{X_t\in B(z_i,w)\}}\right] \\ &= \frac{c^2\Delta^2}{2\sigma^2}\,\mathbb{E}_f\left[N_i\left(\mathcal{A},n\right)\right].\end{aligned}$$

Recall that the term $N_i(\mathcal{A},n)$ denotes the number of times the algorithm $\mathcal{A}$ queries points from the region $B(z_i,w)$ in the first $n$ rounds.

Now, suppose $Z:\Omega\mapsto[0,1]$ be any measurable $[0,1]$ valued random variable. Then by (Garivier et al., 2019, Lemma 1), we get the following result:

$$D_{KL}\left(\mathbb{P}_f,\,\mathbb{P}_{f_i}\right)\geq d_{KL}\left(\mathbb{E}_f\left[Z\right],\,\mathbb{E}_{f'}\left[Z\right]\right),$$

where $d_{KL}(p,q)$ for $p,q\in[0,1]$ denotes the KL-divergence between two Bernoulli random variables with means $p$ and $q$ respectively.

To complete the proof, we select $Z\coloneqq\frac{N_i(\mathcal{A},n)}{n}$, and using the fact (Garivier et al., 2019, Eq. (11)) that $d_{KL}(p,q)\geq -(1-p)\log(1-q)-\log 2$, we get the required inequality

$$\begin{aligned}&\mathbb{E}_f\left[N_i(\tilde{\mathcal{A}},n)\right]\frac{c^2\Delta^2}{2\sigma^2}\geq -\left(1-p_{n,i}\right)\log\left(1-q_{n,i}\right)-\log 2,\quad\text{where}\\ &p_{n,i}=\mathbb{E}_f\left[Z\right],\quad\text{and } q_{n,i}=\mathbb{E}_{f_i}\left[Z\right].\end{aligned}$$

*Remark* 9. The only point at which we exploit the assumption that the observation noise is distributed as $N(0,\sigma^2)$ (i.e., Assumption 2a) is in obtaining the inequality (21). Due to this assumption on the noise, we get a closed form expression for an upper bound on the KL-divergence in (22), i.e., $\left(c^2\Delta^2\right)/2\sigma^2$. In general, if we only assumed that the observation noise was $\sigma^2$ sub-Gaussian, then the same result would hold true with the previous closed-form upper bound replaced by the expression $\sup_{x\in B(z_i,w)}\ D_{KL}\left(\mathbb{P}_{f,Y_n|X_n=x},\,\mathbb{P}_{f_i,Y_n|X_n=x}\right)$.

### B.3. Concluding Theorem 1 from Proposition 3

Theorem 1 follows by repeated application of the result in Proposition 3 to different regions of the input space. More specifically, introduce the following notation:

- As in the previous section, we fix an $a>a_0$, and choose $c=2$ and $\Delta=16n^{-(1-a)}$.
- For $k\geq 0$, set $\mathcal{Z}_k=\{x\in\mathcal{X}:2^k\Delta\leq f(x^*)-f(x)<2^{k+1}\Delta\}$. With $w_k\coloneqq\left((3\times 2^k\Delta M_\nu)/(\lambda M)\right)^{1/\nu}$, we use $m_k$ to denote the $2w_k$ packing number of $\mathcal{Z}_k$.

Since, $f\in\mathcal{H}_{K_\nu}(M)$, we know that the set $\mathcal{Z}_k$ is an empty set for all $k$ larger than a finite value $k_0$. More specifically, we have $k_0=\left\lceil\log_2\left(\frac{\sup_{x\in\mathcal{X}}f(x^*)-f(x)}{\Delta}\right)\right\rceil\leq\left\lceil\log_2\left(\frac{2MK_\nu(0)}{\Delta}\right)\right\rceil$.

Finally, the statement of Theorem 1 follows by $k_0+1$ repeated applications of the intermediate statement proved in Proposition 3 using $\mathcal{Z}=\mathcal{Z}_k$, $\Delta=2^k\Delta$, $c=2$, and $w_k=\left((3\times 2^k\Delta M_\nu)/(\lambda M)\right)^{1/\nu}$ for $k=0,1,\ldots,k_0$.

## C. Proof of Proposition 1

To prove this statement, we appeal to the one-step result obtained in Proposition 3. In particular, we apply Proposition 3 with the following parameters:

- We set $\Delta=\Delta_n=16n^{-(1-a)}$, and $c=c_n\coloneqq\frac{2^{1/d}\bar{c}}{\underline{c}}$ where $\bar{c}$ and $\underline{c}$ are the parameters introduced in Definition 4.
- The set $\mathcal{Z}$ of Proposition 3 now becomes $\{x\in\mathcal{X}:\Delta_n\leq f(x^*)-f(x)<c_n\Delta_n\}$.

- We set the radius of the balls to $w = w_n = \left(\frac{(c_n+1)\Delta_n M_\nu}{M\lambda}\right)^{1/\nu}$, where $\lambda = 1 - \|f\|_{\mathcal{H}_{K_\nu}}/M$, and use $m(\mathcal{Z}, w_n)$ to denote the $2w_n$ packing number of the set $\mathcal{Z}$.

With these parameters, Proposition 3 gives us the following lower bound on the regret:

$$\mathbb{E}\left[\mathcal{R}_n\left(\mathcal{A}, f\right)\right] = \Omega\left(\frac{\sigma^2 m(\mathcal{Z}, w_n)}{\Delta_n c_n^2}\right). \tag{23}$$

To conclude the statement of Proposition 1, we will show that $m = m(\mathcal{Z}, w_n) = \Omega\left(\Delta_n^{\frac{d}{\nu}\left(1-\frac{\nu}{b}\right)}\right)$. First, we introduce the following terms:

$$r_0 \coloneqq \left(\frac{\Delta_n}{\underline{c}}\right)^{1/b}, \qquad r_1 \coloneqq \left(\frac{c_n\Delta_n}{\bar{c}}\right)^{1/b} \quad \text{where} \quad c_n = \frac{2^{1/d}\bar{c}}{\underline{c}} \quad \text{as before.} \tag{24}$$

Next, we use Definition 4 to obtain the following result about $\mathcal{Z}$.

**Lemma 2.** *With $r_0$ and $r_1$ introduced in (24) and $\widetilde{\mathcal{X}}(f, \Delta, c)$ defined above, we have*

$$\mathcal{Z} \supset B(x^*, r_0, r_1) \coloneqq \{x \in \mathcal{X} : r_0 \le \|x - x^*\| < r_1\}$$

*Proof.* Suppose $x \in B(x^*, r_0, r_1)$. Then we have the following:

$$\begin{aligned} \|x - x^*\| \ge r_0 &\coloneqq \left(\frac{\Delta_n}{\underline{c}}\right)^{1/b} \\ \Rightarrow f(x^*) - f(x) &\ge \underline{c} r_0^b \\ \Rightarrow f(x^*) - f(x) &\ge \Delta_n. \end{aligned} \tag{25}$$

Similarly, we also have the following:

$$\begin{aligned} \|x - x^*\| \le r_1 &\coloneqq \left(\frac{c_n\Delta_n}{\bar{c}}\right)^{1/b}, \\ \Rightarrow f(x^*) - f(x) &\le \bar{c} r_1^b \\ \Rightarrow f(x^*) - f(x) &\le \bar{c}\left(\frac{c_n\Delta_n}{\bar{c}}\right) \\ \Rightarrow f(x^*) - f(x) &\le c_n\Delta_n. \end{aligned} \tag{26}$$

Together, (25) and (26) imply that if $x \in B(x^*, r_0, r_1)$ then $x \in \mathcal{Z}$. □

The above statement implies that the $2w_n$ packing number of $\mathcal{Z}$ can be lower-bounded by the $2w_n$ packing number of the smaller set $B(x^*, r_0, r_1)$. Let us denote the $2w_n$-packing number of $B(x^*, r_0, r_1)$ with $\widetilde{m}$. Then, by using the fact that the $2w_n$ packing number is lower bounded by the $2w$ covering number, and employing the standard volume arguments (van Handel, 2014, Lemma 5.13), we conclude that there exists a constant $0 < C_1 < \infty$ such that

$$m(\mathcal{Z}, w_n) \ge m\left(B(x^*, r_0, r_1), w_n\right) \ge C_1\left(\frac{\Delta_n^{1/b}}{\Delta_n^{1/\nu}}\right)^d = C_1\Delta_n^{\frac{d}{\nu}\left(1-\frac{\nu}{b}\right)}.$$

The $\Delta_n^{1/b}$ and the $\Delta_n^{1/\nu}$ terms in the above display arise from the definitions of $r_1 - r_0$ and $w_n$ (defined at the beginning of this proof) respectively. Plugging the lower bound on $m(\mathcal{Z}, w_n)$ back in (23) gives us the required result.

## D. Proof of Theorem 2

We first recall a simple embedding result from Shekhar and Javidi (2020) which is crucial in the adaptive partitioning approach used in our algorithm.

**Fact 1.** *Suppose $f \in \mathcal{H}_{K_\nu}(M)$ with $0 < M < \infty$. Then, we have $|f(x) - f(z)| \leq MC_K\|x - z\|^\xi$ for $\xi \coloneqq \min\{1, \nu\}$ and for some constant $C_K$ depending on $K$ for all $x, z \in \mathcal{X}$.*

Instead of obtaining the bounds on the term $C_K$ in the statement above, we assume that $n$ is large enough to ensure that $C_K \leq \log n$ to simplify the presentation.

We now present the following independence result about the points queried by the algorithm.

**Proposition 4.** *Suppose $\mathcal{P}_t \subset \mathcal{X}_h$ is the active set of points for Algorithm 1 at some time $t$. Let $t_0 < t$ denote the time at which a point from $\mathcal{P}_t$ was first queried, and let $\mathcal{E}_t$ denote the multi-set of points $\{x_{t_0}, \ldots, x_{t-1}\}$ queried by the algorithm. Then the collection of random variables $(y_{t_0}, \ldots, y_{t-1})$ are mutually independent, conditioned on the observations $x_{t_0}, \ldots, x_{t-1}$.*

*Proof.* Using $P$ to represent the joint distribution of the random variables, we proceed as follows:

$$\begin{aligned}
P\left(y_{t_0}, \ldots, y_{t-1} | x_{t_0}, \ldots, x_{t-1}\right) &= \frac{P\left(y_{t_0}, \ldots, y_{t-1}, x_{t_0}, \ldots, x_{t-1}\right)}{P\left(x_{t_0}, \ldots, x_{t-1}\right)} \\
&= \frac{\prod_{s=t_0}^{t-1} P\left(x_s | x_{t_0}, \ldots, x_{s-1}, y_{t_0}, \ldots, y_{s-1}\right) P\left(y_s | x_{t_0}, \ldots, x_s, y_{t_0}, \ldots, y_{s-1}\right)}{P\left(x_{t_0}, \ldots, x_{t-1}\right)} \\
&\overset{(a)}{=} \frac{\prod_{s=t_0}^{t-1} P\left(x_s | x_{t_0}, \ldots, x_{s-1}\right) P\left(y_s | x_s\right)}{P\left(x_{t_0}, \ldots, x_{t-1}\right)} \\
&= \frac{P\left(x_{t_0}, \ldots x_{t-1}\right) \left(\prod_{s=t_0}^{t-1} P\left(y_s | x_s\right)\right)}{P\left(x_{t_0}, \ldots, x_{t-1}\right)} \\
&\overset{(b)}{=} \prod_{s=t_0}^{t-1} P\left(y_s | x_{t_0}, \ldots, x_{t-1}\right).
\end{aligned}$$

In the above display,
**(a)** uses the fact that at any $s \geq t_0$, the query point $x_s$ only depends on the previous query points $x_{t_0}, \ldots, x_{s-1}$ and not on the observations; and the fact that conditioned on $x_s$, the observation $y_s$ is independent of the query points $x_{t_0}, \ldots, x_{s-1}$ and observations $y_{t_0}, \ldots, y_{s-1}$.
**(b)** uses the fact that conditioned on $x_s$ the observation $y_s$ is independent of $x_{t_0}, \ldots, x_{s-1}, x_{s+1}, \ldots, x_{t-1}$.

The equality **(b)** implies the conditional independence of the observations given the query points belonging to the current active set $\mathcal{P}_t$, as required. □

Having obtained the conditional independence property of the query points, we now present the key concentration result that leads to the required regret bounds.

**Lemma 3.** *For some $t \geq 1$, let $\mathcal{P}_t$, $t_0$ and $\mathcal{E}_t$ be the same as in Proposition 4. Then, for a given $\delta \in (0, 1)$, the following is true:*

$$P\left(\exists x \in \mathcal{P}_t, \text{ s.t. } |f(x) - \mu_t(x)| > \beta_t \sigma_t(x)\right) \leq \delta_t \coloneqq \frac{6\delta}{t^2\pi^2},$$
$$\text{where } \beta_t = \sqrt{2\sigma^2 \log\left(\frac{|\mathcal{P}_t|\pi^2 t^2}{3\delta}\right)}$$

*Recall that the terms $\mu_t(\cdot)$ and $\sigma_t(\cdot)$ represent the posterior mean and standard-deviation functions computed by the subroutine* `ComputePosterior` *introduced in Definition 6.*

*Proof.* To prove this result, we rely on the following facts derived by Valko et al. (2013) while proving their Lemma 2. For $x \in \mathcal{P}_t$, there exists $\alpha_{t_0}, \ldots, \alpha_{t-1} \in \mathbb{R}$ (depending on $x$) such that the following holds:

$$f(x) - \mu_t(x) = \sum_{i=t_0}^{t-1} \alpha_i \left(y_i - f(x_i)\right) + B_x$$
$$\text{with } \sum_{i=t_0}^{t-1} \alpha_i^2 \le \sigma_t(x)^2 \quad \text{and} \quad |B_x| \le \tau^{-1/2} M \sigma_t(x),$$

for all $x \in \mathcal{P}_t$. Recall that $\tau$ is the regularization parameter used in the subroutine `ComputePosterior`, while $M$ is the upper bound on the RKHS norm of $f$.

Now, we use the conditional independence property derived in Proposition 4 along with the conditional $\sigma^2$-sub-Gaussianity of the observation noise to get the required concentration result.

In particular, for a given $t \ge 1$ and a fixed $x \in \mathcal{P}_t$, we have

$$\begin{aligned}
P\left(f(x) - \mu_t(x) > \beta_t \sigma_t(x)\right) &= \mathbb{E}\left[P\left(\sum_{i=t_0}^{t-1} \alpha_i \left(y_i - f(x_i)\right) > \beta_t \sigma_t(x) | \left(x_i\right)_{i=t_0}^{t-1}\right)\right] \\
&\le \mathbb{E}\left[\mathbb{E}\left[\exp\left(\lambda \sum_{i=t_0}^{t-1} \alpha_i (y_i - f(x_i))\right) e^{-\lambda \beta_t \sigma_t(x)} \mid \left(x_i\right)_{i=t_0}^{t-1}\right]\right] && (27) \\
&= \mathbb{E}\left[\prod_{i=t_0}^{t-1} \mathbb{E}\left[\exp\left(\lambda \alpha_i \left(y_i - f(x_i)\right)\right) | \left(x_i\right)_{i=t_0}^{t-1}\right]\right] && (28) \\
&\le \exp\left(\frac{\lambda^2 \sigma^2}{2} \sum_{i=t_0}^{t-1} \alpha_i^2 - \lambda \beta_t \sigma_t(x)\right) && (29) \\
&\le \exp\left(\frac{\lambda^2 \sigma^2}{2} \sigma_t(x)^2 - \lambda \beta_t \sigma_t(x)\right) && (30) \\
&= \exp\left(-\frac{\beta_t^2}{2\sigma^2}\right). && (31)
\end{aligned}$$

In the above display,
(27) follows by an application of Chernoff's inequality with some constant $\lambda > 0$ to be selected later,
(28) follows from the conditional independence property derived in Proposition 4,
(29) uses the fact that, conditioned on $x_i$, the random variable $y_i - f(x_i)$ is zero-mean $\sigma^2$ sub-Gaussian,
(30) uses the fact that $\sum_{i=t_0}^{t-1} \alpha_i^2 \le \sigma_t(x)^2$, and
(31) follows by selecting $\lambda = \beta_t / (\sigma^2 \sigma_t(x))$.

Repeating the argument of the previous display with $\mu_t(x) - f(x)$, in the place of $f(x) - \mu_t(x)$, gives us that

$$P\left(|f(x) - \mu_t(x)| > \beta_t \sigma_t(x)\right) \le 2 \exp\left(-\frac{\beta_t^2}{2\sigma^2}\right).$$

Next, using the fact that $\beta_t > \sqrt{2\sigma^2 \log\left(\frac{|\mathcal{P}_t| \pi^2 t^2}{3\delta}\right)}$, we have by a union bound over $x \in \mathcal{P}_t$:

$$\begin{aligned}
P\left(\exists x \in \mathcal{P}_t, \text{ s.t. } |f(x) - \mu_t(x)| > \beta_t \sigma_t(x)\right) &\le \sum_{x \in \mathcal{P}_t} 2 \exp\left(-\frac{\beta_t^2}{2\sigma^2}\right) \\
&< 2 \exp\left(-\log\left(\frac{|\mathcal{P}_t| \pi^2 t^2}{3\delta}\right)\right) \\
&= \frac{6\delta}{\pi^2 t^2} \coloneqq \delta_t.
\end{aligned}$$

□

Since $\sum_{t=1}^{n} \delta_t < \delta$, we note that the following event $\mathcal{E}$, occurs with probability at least $1-\delta$

$$\mathcal{E} := \cap_{t=1}^{n} \cap_{x \in \mathcal{P}_t} \{|f(x) - \mu_t(x)| \leq \beta_t \sigma_t(x)\}. \tag{32}$$

Throughout the rest of the proof, we will work under the event $\mathcal{E}$ with $\delta = 1/n$. Hence, the expected regret of the algorithm $\mathcal{A}_1$ can then be upper bounded by

$$\begin{aligned}
\mathbb{E}[\mathcal{R}_n(\mathcal{A}_1, f)] &= \mathbb{E}[\mathcal{R}_n(\mathcal{A}_1, f)\mathbb{1}_{\{\mathcal{E}\}}] + \mathbb{E}[\mathcal{R}_n(\mathcal{A}_1, f)\mathbb{1}_{\{\mathcal{E}^c\}}] \\
&\leq \mathbb{E}[\mathcal{R}_n(\mathcal{A}_1, f)\mathbb{1}_{\{\mathcal{E}\}}] + \mathbb{P}\left(\mathcal{E}^c\right) \times n \times \sup_{x \in \mathcal{X}} (f(x^*) - f(x)) \\
&\leq \mathbb{E}[\mathcal{R}_n(\mathcal{A}_1, f)\mathbb{1}_{\{\mathcal{E}\}}] + 2\|f\|_{\mathcal{H}_{K_\nu}} \text{diam}(\mathcal{X}) \\
&\leq \mathbb{E}[\mathcal{R}_n(\mathcal{A}_1, f)\mathbb{1}_{\{\mathcal{E}\}}] + \mathcal{O}(1).
\end{aligned} \tag{33}$$

Since the second term is upper bounded by a constant, it suffices to show that the required upper bounds hold for the regret incurred under the event $\mathcal{E}$.

We now obtain a result about the sub-optimality of the points queried by the algorithm.

**Lemma 4.** *Suppose event $\mathcal{E}$ introduced in (32) occurs, and the algorithm queries a point $x_t \in \mathcal{P}_t \subset \mathcal{X}_h$ for some $h \geq 1$. Then we have*

$$f(x^*) - f(x_t) \leq \left(7Lv_1^{\xi}\rho^{-\xi}\right)\rho^{h\xi} = \mathcal{O}\left(\rho^{h\xi}\right).$$

*Proof.* Since $h \geq 1$, the set $\mathcal{P}_t$ must have been formed by a call to the `RefinePartition` subroutine. Let $x_t = x_{h,i}$ for some $i \in \{1, \ldots, 2^h\}$ and furthermore, denote its parent node by $x_{h',i'}$ where $h' = h-1$ and $i' = \lceil i/2 \rceil$. Assume that the active set $\mathcal{P}_t$ was formed by a call to the `RefinePartition` subroutine at some time $t_0 < t$. Then the following must be true:

$$\begin{aligned}
f(x_{h,i}) = f(x_t) &\geq f(x_{h',i'}) - 2\overbrace{L(v_1\rho^{h'})^{\xi}}^{:=V_{h'}} \geq \mu_{t_0}(x_{h',i'}) - 2\beta_{t_0}\sigma_{t_0}(x_{h',i'}) - V_{h'} && (34)\\
&\geq \mu_{t_0}(x_{h',i'}) + \beta_{t_0}\sigma_{t_0}(x_{h',i'}) - 3V_{h'} && (35)\\
&\geq \max_{x \in \mathcal{P}_{t_0}} \left(\mu_{t_0}(x) - \beta_{t_0}\sigma_{t_0}(x)\right) - 3V_{h'} && (36)\\
&\geq (f(x^*) - 4V_{h'}) - 3V_{h'} = f(x^*) - 7V_{h'}. && (37)
\end{aligned}$$

In the above display,

- the first inequality in (34) uses the fact that $f$ is $(L, \xi)$ Hölder continuous, while that second inequality uses the fact that under the event $\mathcal{E}$, we have $f(x_{h',i'}) \geq \mu_{t_0}(x_{h',i'}) - \beta_{t_0}\sigma_{t_0}(x_{h',i'})$,
- (35) follows by adding and subtracting $2\beta_{t_0}\sigma_{t_0}(x_{h',i'})$, and then using the fact that $2\beta_{t_0}\sigma_{t_0}(x_{h',i'})$ must be smaller than $2V_{h'} := L(v_1\rho^{h'})^{\xi}$, due to line 6 in Algorithm 1,
- (36) then uses the fact $\mu_{t_0}(x_{h',i'}) + \beta_{t_0}\sigma_{t_0}(x_{h',i'})$ must be larger than the highest lower bound, $\max_{x \in \mathcal{P}_{t_0}} \mu_{t_0}(x) - \beta_{t_0}\sigma_{t_0}(x)$ in order for $x_{h,i}$ to be included in the updated $\mathcal{P}_{t_0}$ returned by `RefinePartition`,
- and finally, (37) uses the fact that $f(x^*) - 2V_{h'} \geq \max_{x \in \mathcal{P}_{t_0}} \mu_{t_0}(x) - \beta_{t_0}\sigma_{t_0}(x)$. To see this, suppose $x_{h',j}$ denotes the point in $\mathcal{P}_{t_0}$ such that $x^* \in \mathcal{X}_{h',j}$. Then we must have the following:

$$\begin{aligned}
\max_{x \in \mathcal{P}_{t_0}} \mu_{t_0}(x) - \beta_{t_0}\sigma_{t_0}(x) \geq& \mu_{t_0}\left(x_{h',j}\right) - \beta_{t_0}\sigma_{t_0}\left(x_{h',j}\right) \geq f(x_{h',j}) - 2\beta_{t_0}(x_{h',j}) \\
\geq& f(x^*) - 2V_{h'} - 2\beta_{t_0}(x_{h',j}) \geq f(x^*) - 4V_{h'}.
\end{aligned}$$

□

The previous lemma gives us a bound on the suboptimality of any point queried by the algorithm at a time $t$ in terms of the parameter $\rho$ and the depth of the cell $h$. Now, let $N_h$ denote the number of times the algorithm queries a point at level $h$, i.e., lying in the subset $\mathcal{X}_h$. Then we have the following regret decomposition (assuming the event $\mathcal{E}$ defined in (32) occurs):

$$\mathcal{R}_n(\mathcal{A}_1, f) = \sum_{t=1}^{n} f(x^*) - f(x_t) = \mathcal{O}\left(\sum_{h\geq 0} N_h \rho^h\right).$$

To complete the proof, it remains to get an upper bound on the term $N_h$, which we do in two ways: one in terms of the maximum information gain $\gamma_n$, and the other in terms of the upper-complexity term $\overline{\mathcal{C}}_f$ introduced in Definition 8. The former will imply the minimax near-optimality of $\mathcal{A}_1$, while the latter will lead to improved regret (beyond the minimax value) on some easier problem instances.

We now present the $\gamma_n$ based upper bound on $N_h$. As an immediate consequence of this bound, we also observe that $\mathcal{A}_1$ is minimax near-optimal.

**Lemma 5.** *The number of queries made by $\mathcal{A}_1$ at level $h$ of the tree satisfies $N_h = \widetilde{\mathcal{O}}\left(\rho^{-2h\xi}\gamma_n\right)$. As a consequence of this, we obtain the following upper bound on the regret:*

$$\mathbb{E}\left[\mathcal{R}_n(\mathcal{A}_1, f)\right] = \widetilde{\mathcal{O}}\left(\sqrt{n\gamma_n}\right) = \widetilde{\mathcal{O}}\left(n^{a^*_\nu}\right), \quad \textit{with} \quad a^*_\nu = \frac{\nu + d}{2\nu + d}.$$

*Proof.* This result follows by using (Valko et al., 2013, Lemma 4 and 5) to get that $N_h = \widetilde{\mathcal{O}}\left(\rho^{-h\xi}\beta_n\sqrt{\gamma_n N_h}\right)$. Dividing both sides by $\sqrt{N_h}$ and taking the square gives the upper bound on $N_h$.

Next, under event $\mathcal{E}$, we have $\mathcal{R}_n = \mathcal{O}\left(\sum_{h=0}^{h_{\max}} \rho^{h\xi} N_h\right) = \widetilde{\mathcal{O}}\left(\sum_{h=0}^{h_{\max}} \beta_n\sqrt{\gamma_n N_h}\right) = \widetilde{\mathcal{O}}\left(h_{\max}\sqrt{\gamma_n n}\right) = \widetilde{\mathcal{O}}\left(\sqrt{\gamma_n n}\right)$. In the last two equalities, we used the fact that $\beta_n = \mathcal{O}\left(\sqrt{\log n}\right)$ and $h_{\max} = \log n$, and hence are absorbed by the hidden polylogarithmic leading constant in the notation $\widetilde{\mathcal{O}}(\cdot)$.

□

**Lemma 6.** *Assume that the event $\mathcal{E}$ holds, and let $N_h$ denote the number of queries made by $\mathcal{A}_1$ at level $h$. Introduce the set $W_h \coloneqq \{x \in \mathcal{X} : f(x^*) - f(x) \leq (7Lv_1^\xi\rho^{-\xi})\rho^{h\xi}\}$, and let $m_h = m\left(W_h, 2v_2\rho^h\right)$ denote the $2v_2\rho^h$-packing number of the set $W_h$. Then, $N_h$ is upper bounded by $\widetilde{\mathcal{O}}\left(\rho^{-2h\xi}m_h\right)$.*

*Proof.* Suppose a point $x_{h,i} \in \mathcal{X}_h$ is evaluated $n_{h,i}$ times before a call to `RefinePartition` is made. Then we must have that

$$n_{h,i} = \left\lceil \frac{\beta_n^2\tau^2}{L^2 v_1^{2\xi}\rho^{2h\xi}} \right\rceil. \tag{38}$$

This is due to the following fact: suppose that at time $t$, the point $x_{h,i}$ has been evaluated $s$ times. Then by (Shekhar and Javidi, 2018, Proposition 3) we know that the posterior standard deviation at $x_{h,i}$ must satisfy $\sigma_t(x_{h,i}) \leq \tau/\sqrt{s}$. Plugging $s \leftarrow n_{h,i}$ from (38) in this bound implies that after $n_{h,i}$ evaluations, the condition $\beta_t\sigma_t(x_{h,i}) < L(v_1\rho^h)^\xi$ is satisfied, and hence the point $x_{h,i}$ will not be evaluated anymore. Furthermore, we also know that if $\mathcal{P}_t \subset \mathcal{X}_h$, then it also satisfies the following two properties: **(i)** $\mathcal{P}_t \subset \mathcal{Z}_h \coloneqq \left\{x \in \mathcal{X} : f(x^*) - f(x) \leq \left(7Lv_1^\xi\rho^{-\xi}\right)\rho^{h\xi}\right\}$, and **(ii)** any two points in $\mathcal{P}_t$ are separated by a distance of $2v_2\rho^h$. Together, these two facts imply that $\mathcal{P}_t$ must be a packing set of $\mathcal{Z}_h$, and thus we can upper bound $|\mathcal{P}_t|$ with $m_h$, the $2v_2\rho^h$ packing number of $\mathcal{Z}_h$. As a consequence, we have $N_h \leq n_{h,i}m_h = \widetilde{\mathcal{O}}\left(\rho^{-2h\xi}m_h\right)$. □

It remains to show that the expected regret of $\mathcal{A}_1$ is upper bounded by the upper-complexity term $\overline{\mathcal{C}}_f$.

**Lemma 7.** *Introduce the term $H_n = \max\{H : \sum_{h=0}^{H} \rho^{-2h\xi}m_h \leq n\}$, and define $\Delta_n = \min\{n^{-(1-a^*_\nu)}, \rho^{H_n\xi}\}$. Then, we have*

$$\mathbb{E}[\mathcal{R}_n\left(\mathcal{A}_1, f\right)] = \widetilde{\mathcal{O}}\left(\overline{\mathcal{C}}_f(\Delta_n)\right).$$

*Proof.* We will work under the event $\mathcal{E}$ introduced in (32), that occurs with probability at least $1 - 1/n$. The proof of this result follows by employing the upper bound on $N_h$ derived in Lemma 6, and rearranging the resulting terms to form the upper-complexity term.

In particular, we note that $\mathcal{R}_n(\mathcal{A}_1, f) \leq \sum_{h=0}^{H_n} \widetilde{\mathcal{O}}\left(\rho^{h\xi} N_h\right)$. Now, due to the upper bound on $N_h$ obtained in Lemma 6, we have $\mathcal{R}_n(\mathcal{A}_1, f) = \widetilde{\mathcal{O}}\left(\sum_{h=0}^{H_n} \rho^{-h\xi} m_h\right)$. To rewrite this in terms of $\overline{\mathcal{C}}_f$, note that $\Delta_n \leq \rho^{H_n \xi}$, and for $k \geq 0$, define

$$\begin{aligned}
\widetilde{\mathcal{Z}}_k = W_{H_n - k} &= \{x : f(x^*) - f(x) \leq (7Lv_1^{\xi}\rho^{-\xi})\rho^{(H_n-k)\xi}\} \\
&= \{x : f(x^*) - f(x) \leq (7Lv_1^{\xi}\rho^{-\xi})\rho^{-\xi k}\rho^{H_n\xi}\} \\
&= \{x : f(x^*) - f(x) \leq (7Lv_1^{\xi}\rho^{-\xi})\rho^{-\xi k}\Delta_n\}.
\end{aligned}$$

Having defined $\widetilde{\mathcal{Z}}_k$, we note that the term $\widetilde{m}_k$ in the definition of $\overline{\mathcal{C}}_f$ is the same as $m_{H_n - h} = m\left(\widetilde{\mathcal{Z}}_k, v_2\rho^{H_n-k}\right)$ : the $2v_2\rho^{H_n-k}$-packing number of $\widetilde{\mathcal{Z}}_k$. Finally, noting that $\rho^{H_n-k}\xi = (\rho^{-\xi k})\Delta_n$, we have the following:

$$\sum_{h=0}^{H_n} \rho^{-\xi h} m_h = \sum_{k=0}^{H_n} \rho^{-\xi(H_n-k)}\widetilde{m}_k = \sum_{k=0}^{H_n} \frac{\widetilde{m}_k}{(1/\rho^{\xi})^k \Delta_n} = \overline{\mathcal{C}}_f(\Delta_n).$$

This completes the proof. □

## E. Proof of Proposition 2

As shown in (33), it suffices to get the bound on the regret under the $1 - 1/n$ probability event $\mathcal{E}$ introduced in (32). When the objective function, $f$, satisfies the local growth condition with exponent $b$, we can show that the regions $W_h = \{x : f(x^*) - f(x) \leq (7Lv_1^{\xi}\rho^{-\xi})\rho^{h\xi}\}$ is contained in a ball centered at the optimal point $x^*$. In particular, due to the local-growth condition, it follows that $W_h \subset B\left(x^*, c'\rho^{h\xi/b}\right)$ for some constant $c'$. As $m_h = m\left(W_h, v_2\rho^h\right)$ is the $2v_2\rho^h$-packing number of the set $W_h$, we can bound it from above by using volume arguments to get that $m_h = \left(\rho^{hd(1-\xi/b)}\right)$.

Combining this with the result of Lemma 6, we get that

$$\mathcal{R}_n(\mathcal{A}_1, f) = \widetilde{\mathcal{O}}\left(\sum_{h=0}^{h_{\max}} \rho^{-h\xi}\rho^{-hd(1-\xi/b)}\right).$$

Introducing the term $\widetilde{d} \coloneqq d(1-\xi/b)$, we have $\mathcal{R}_n(\mathcal{A}_1, f) = \widetilde{\mathcal{O}}\left(\sum_{h=0}^{h_{\max}} \rho^{-h(\xi+\widetilde{d})}\right)$. Proceeding as in the proof of Lemma 7, introduce the term $\widetilde{H}_n = \max\{H \leq h_{\max} : \sum_{h=0}^{H} \rho^{-h(2\xi+\widetilde{d})}\}$, we see that $\rho^{-\widetilde{H_n}} = \widetilde{\mathcal{O}}\left(n^{1/(2\xi+\widetilde{d})}\right)$. This gives us the following:

$$\mathcal{R}_n(\mathcal{A}_1, f) = \widetilde{\mathcal{O}}\left(\sum_{h=0}^{H_n} \rho^{-h(\xi+\widetilde{d})} + \rho^{H_n\xi} n\right) = \widetilde{\mathcal{O}}\left(n^{(\xi+\widetilde{d}))/(2\xi+\widetilde{d})}\right).$$

## F. Extension to Lipschitz Bandits

We note that the ideas used in obtaining the instance-dependent lower bound in Theorem 1 can also be applied to related problems, such as Lipschitz bandits and kernelized level-set estimation. In this section, we state the analogous result for the Lipschitz bandit problem. Here, the goal is to design an adaptive querying strategy to optimize an unknown Lipschitz continuous objective function $f$ with a Lipschitz constant bounded above by $L$, via noisy zeroth-order queries. For this problem, we can prove an analog of Theorem 1.

**Definition 18.** Let $f$ be a $(1-\lambda)L$-Lipschitz function for some $\lambda \in (0,1)$. Fix a $\Delta > 0$, and introduce the set $\mathcal{Z}_k \coloneqq \{x \in \mathcal{X} : 2^k\Delta \leq f(x^*) - f(x) < 2^{k+1}\Delta\}$. Introduce the radius $w_k = 3 \times 2^k\Delta/(\lambda L)$, and let $m_k$ denote the $2w_k$ packing number of the set $\mathcal{Z}_k$ for $k \geq 0$. Then, we can define the following complexity term:

$$\mathcal{C}_f^{(Lip)}(\Delta, L, \lambda) \coloneqq \sum_{k \geq 0} \frac{m_k}{2^{k+2}\Delta} \geq \frac{m_0}{4\Delta}.$$

Then, proceeding as in the proof of Theorem 1, we can obtain the following lower bound.

**Proposition 5.** *For a $(1-\lambda)L$-Lipschitz function $f$, the expected regret of an $a_0$-consistent (for the family of $L$-Lipschitz functions) algorithm $\mathcal{A}$ satisfies the following for any $a > a_0$:*

$$\mathbb{E}\left[\mathcal{R}_n\left(\mathcal{A}, f\right)\right] = \Omega\left(\sigma^2 \mathcal{C}_f^{(Lip)}\left(n^{-(1-a)}, L, \lambda\right)\right).$$

The general steps involved in obtaining this statement are similar to those used in the proof of Theorem 1, and we omit the details. In particular, the main difference is that we now use the bump function of the form $g(x) = \max\{0, L(1-\|x\|)\}$. We can check that this function is $L$-Lipschitz and supported on the unit ball.

## G. Additional Figures

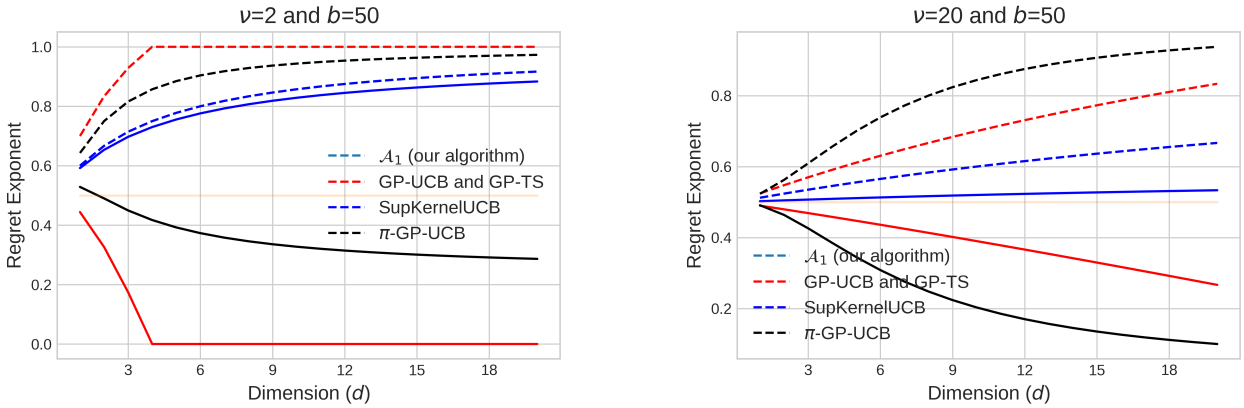


*Figure 3.* The two figures plot the variation of the exponent of different regret upper and lower bounds (i.e., $\alpha$, if regret $\approx n^{\alpha}$) with dimension, on a problem instance $f \in \mathcal{H}_{K_\nu}$, satisfying $f \approx (x - x^*)^b$ near its optimum.