No-Regret Algorithms for Safe Bayesian Optimization with Monotonicity Constraints

Arpan Losalka
National University of Singapore

Jonathan Scarlett
National University of Singapore

Abstract

We consider the problem of sequentially maximizing an unknown function f over a set of actions of the form $(s, \mathbf{x})$, where the selected actions must satisfy a safety constraint with respect to an unknown safety function g. We model f and g as lying in a reproducing kernel Hilbert space (RKHS), which facilitates the use of Gaussian process methods. While existing works for this setting have provided algorithms that are guaranteed to identify a near-optimal safe action, the problem of attaining low cumulative regret has remained largely unexplored, with a key challenge being that expanding the safe region can incur high regret. To address this challenge, we show that if g is monotone with respect to just the *single variable* s (with no such constraint on f), sublinear regret becomes achievable with our proposed algorithm. In addition, we show that a modified version of our algorithm is able to attain sublinear regret (for suitably defined notions of regret) for the task of finding a near-optimal s corresponding to every $\mathbf{x}$, as opposed to only finding the global safe optimum. Our findings are supported with empirical evaluations on various objective and safety functions.

1 INTRODUCTION

Sequential optimization of an unknown function is an important task with many applications, and comes with an interesting set of challenges in the scenario that function queries are expensive. Bayesian optimization is a popular approach for this task, with applications including robotics (Lizotte et al., 2007), environmental monitoring (Srinivas et al., 2012), adaptive clinical trial design (Takahashi and Suzuki, 2021), hyperparameter tuning in machine learning (Snoek et al., 2012) and recommendation systems (Vanchinathan et al., 2014), among others.

This black-box optimization problem becomes even more challenging when considering a safety constraint along with the optimization objective. One important notion of safety that has been considered in the literature is to only allow actions for which an unknown safety function takes values above a pre-defined safety threshold (Sui et al., 2015). Various methods have been proposed for this task, such as SafeOpt (Sui et al., 2015), StageOpt (Sui et al., 2018) and SafeOpt-MC (Berkenkamp et al., 2021), among others. The core idea of these algorithms is to start from a safe seed set of inputs, and cautiously expand the set and identify the most promising regions within it, in order to reach the optimal action. Accordingly, the main performance metrics considered have been *expanding the safe set as much as possible* and *returning a single near-optimal action*. In contrast, the goal of small *cumulative regret* has remained relatively unexplored in safe settings, despite being the most widely-adopted performance measure in the vanilla setting. A key difficulty in the safe setting is that expanding the set of known safe points is "purely explorative" and may require sampling many highly suboptimal points.

In this work, we consider the scenario where the safety function g is known to increase monotonically with respect to a safety variable $s \in \mathcal{D}_{\mathcal{S}}$, while the objective function f is distinct from g and need not have any such structure. We show that with this mild assumption, strong guarantees on the cumulative regret become possible. We also introduce other notions of regret associated with finding the best s for *every* $\mathbf{x} \in \mathcal{D}_{\mathcal{X}}$ *separately*, and we show that our algorithms can be simplified (while maintaining similar guarantees) when *both* f and g are monotone in s.

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

Motivating Applications: Consider the problem of an adaptive Phase I/II clinical trial design for the purpose of finding drug doses that simultaneously satisfy safety constraints with respect to drug toxicity, as well as achieve optimal efficacy (Berry, 2012). A common structure exhibited by the toxicity of various classes of drugs is that it increases monotonically as a function of the drug dosage (Chevret, 2006). While the efficacy may also increase similarly, it is not the case in general, especially when drug combinations are used (Cai et al., 2014). In such a scenario, the problem of finding the optimal safe drug dose fits our problem formulation, since both toxicity (g) and efficacy (f) are unknown functions of the drug doses (with s denoting dosage of one drug, and $\mathbf{x}$ denoting that of other drugs). Here, we need to optimize f while satisfying the toxicity threshold $g(s, \mathbf{x}) \leq h$, and regret minimization in this setting implies that we not only find the optimum (minimizing simple regret), but we also maximize benefits to the trial participants simultaneously (minimizing cumulative regret).

The requirement of a trial may also be to find the optimal dose for a range of patient's characteristics (e.g., age group, gender, etc.). In this case, our alternate problem setting is relevant, where we want to find the optimum s (dose) for every $\mathbf{x}$ (patient's characteristics), while satisfying safety constraints (toxicity threshold). The problem formulation also extends naturally when considering drug-combinations, as long as monotonic behavior of drug toxicity holds with respect to *at least one drug dose*.

Another application area suited to the goal of choosing parameters to optimize performance (with respect to f) while ensuring safety (with respect to g) is robotics (Berkenkamp et al., 2017). Here, the notion of a "safety variable" s might be explicitly incorporated into the system by design, i.e., we have a controllable variable that directly dictates "how cautiously" the task is performed. Alternatively, certain variables such as acceleration and torque might be *implicitly* connected to safety and naturally provide our required monotonicity constraint.

Related Work: The problem of safe Bayesian optimization was first considered by Sui et al. (2015), who proposed the SafeOpt algorithm for this task. This algorithm, as well as other algorithms that were proposed subsequently such as StageOpt (Sui et al., 2018), SafeOpt-MC (Berkenkamp et al., 2021), and GOOSE (Turchetta et al., 2019), aims to expand a safe seed set widely enough to guarantee identifying a near-optimal safe point (excluding those from regions that are "unreachable"). Extensions have also been provided to reinforcement learning (Turchetta et al., 2016; Berkenkamp et al., 2017; Turchetta et al., 2020).

A distinct approach that can give low cumulative regret was proposed in (Amani et al., 2021), but since it expands the safe set in a "one-shot" manner, it is only suited to kernels that are finite-dimensional or extremely smooth. Improvements over the above algorithms are also possible for safety functions modeled by dynamical systems (Baumann et al., 2021; Sukhija et al., 2022), but our focus is on static functions. We refer the reader to (Losalka and Scarlett, 2023) for further discussion on all of these works.

The closest work to ours is that of Losalka and Scarlett (2023), who also use monotonicity of the unknown function f with respect to a safety variable s, and design the M-SafeUCB algorithm. While they prove a sublinear regret bound, the applicability of the algorithm is limited by the fact that the safety constraint is defined with respect to f itself. In the more general setting where f and g are distinct, their algorithm is only directly suitable when *both f and g* are monotone in s (see Section 3 for further details). A naive strategy for overcoming this would be to explore using g and then pass the resulting safe set to an optimizer for f, but the former step may already incur high cumulative regret with respect to f. Overall, our more general setting significantly complicates both the algorithm design and the theoretical analysis.

Contributions: Our main contributions are summarized as follows.

1. We introduce the problem of safe optimization of an unknown function f when the safety function g is known to be monotone with respect to a safety variable s. We propose an algorithm, M-SafeOpt, which uses a novel acquisition function designed to balance exploration and exploitation, along with the elimination of provably suboptimal $\mathbf{x}$'s, to attain sublinear cumulative regret.
2. We consider an alternative setting where the goal is to find the optimal action corresponding to *every* $\mathbf{x} \in \mathcal{D}_{\mathcal{X}}$ (i.e., the best safe s for every $\mathbf{x}$), and adapt our algorithm for this task. We provide modified cumulative regret notions capturing both the degree of optimality of sampled points and a *worst-case* notion over all $\mathbf{x}$ (see Section 2 for details), with both guaranteed to be sublinear.
3. We show how our algorithm can be simplified in the scenario that both f and g increase monotonically with respect to s, while attaining the same theoretical guarantees for the goal of safe optimization across the input domain. Alternatively, when the goal is to optimize s for every $\mathbf{x}$, we show how further simplification recovers the M-

SafeUCB algorithm and its guarantees.

4. We empirically evaluate our proposed algorithms on various synthetic functions, showing significant improvements over several natural baselines.

2 PROBLEM STATEMENT

We consider the problem of sequentially maximizing an unknown function $f : \mathcal{D} \to \mathbb{R}$ over a set of actions $\mathcal{D} = \mathcal{D}_\mathcal{S} \times \mathcal{D}_\mathcal{X}$ while satisfying a given safety constraint with respect to another unknown function $g : \mathcal{D} \to \mathbb{R}$, where $\mathcal{D}_\mathcal{X} \subset \mathbb{R}^d$ is a compact set and $\mathcal{D}_\mathcal{S} = [0, 1]$. The function g is assumed to increase monotonically in the first argument $s \in \mathcal{D}_\mathcal{S}$.

At each round t, the algorithm selects an action $(s_t, \mathbf{x}_t) \in \mathcal{D}_\mathcal{S} \times \mathcal{D}_\mathcal{X}$, and subsequently observes a noisy evaluation of the objective function $y_t^f = f(s_t, \mathbf{x}_t) + \epsilon_t^f$, as well as that of the safety function, $y_t^g = g(s_t, \mathbf{x}_t) + \epsilon_t^g$. At round t, the selected action is a function of the history $\mathcal{H}_{t-1} = \{(s_k, \mathbf{x}_k, y_k^f, y_k^g) : k = 1, \ldots, t-1\}$, and is required to satisfy the safety condition[1] $g(s_t, \mathbf{x}_t) \leq h$ as formalized below.

Goal: We consider two distinct objectives: (i) finding the global safe optimum, or (ii) finding the optimal f-value for every $\mathbf{x} \in \mathcal{D}_\mathcal{X}$. In both cases, an algorithm must satisfy the safety constraint: (iii) $g(s_t, \mathbf{x}_t) \leq h \; \forall t \geq 1$ with high probability.

When the goal is to sequentially maximize f over $\mathcal{D}$ (goal (i)), we consider the following definition of cumulative regret:

$$R_T = \sum_{t=1}^{T} r_t, \text{ with } r_t = f(s_*, \mathbf{x}_*) - f(s_t, \mathbf{x}_t), \quad (1)$$

where $(s_*, \mathbf{x}_*) = \arg\max_{(s,\mathbf{x}) \in \mathcal{D} : g(s,\mathbf{x}) \leq h} f(s, \mathbf{x})$ is an optimal safe action. This matches the standard cumulative regret notion in black-box optimization, but restricted to safe actions.

When the goal is to find the optimal f-value for every $\mathbf{x} \in \mathcal{D}_\mathcal{X}$, we consider the following modified definition:

$$R'_T = \sum_{t=1}^{T} r'_t, \text{ with } r'_t = f(s_*^{(\mathbf{x}_t)}, \mathbf{x}_t) - f(s_t, \mathbf{x}_t), \quad (2)$$

where $s_*^{(\mathbf{x})} = \arg\max_{s \in \mathcal{D}_\mathcal{S} : g(s,\mathbf{x}) \leq h} f(s, \mathbf{x})$ denotes the optimal safe s given $\mathbf{x}$. Note that this definition varies from the usual notion of regret used in bandit problems. We use this formulation here to evaluate whether an algorithm achieves sublinear regret with respect to whichever $(s_t, \mathbf{x}_t)$ it chooses in each round t. However, minimizing this quantity alone would not result in a complete evaluation of the algorithm for this goal. This is because, for example, it may choose to select the same $\mathbf{x}$ in every round, and minimize regret only with respect to this $\mathbf{x}$. Given that we want the algorithm to simultaneously find the maximum f-value for every $\mathbf{x}$ for goal (ii), we additionally specify the following quantity that we also seek to minimize:

$$R_T^\mathcal{X} = \sum_{t=1}^{T} r_t^\mathcal{X}, \text{ with } r_t^\mathcal{X} = \max_{\mathbf{x} \in \mathcal{D}_\mathcal{X}} \Big(f(s_*^{(\mathbf{x})}, \mathbf{x}) - f(\hat{s}_t^{(\mathbf{x})}, \mathbf{x}) \Big), \quad (3)$$

where $\hat{s}_t^{(\mathbf{x})}$ denotes the algorithm's "best guess" of the optimal safe $s \in \mathcal{D}_\mathcal{S}$ for a given $\mathbf{x}$ after round t (see Section 3 for our specific choice of $\hat{s}_t^{(\mathbf{x})}$).

While minimizing R'_T ensures that the algorithm makes progressively better choices, minimizing $R_T^\mathcal{X}$ ensures that for every $\mathbf{x} \in \mathcal{D}_\mathcal{X}$, the f-values of the best actions estimated by the algorithm get progressively closer to the true optima. Note that neither of these two objectives implies the other, since one specifically concerns the *actions selected* (R'_T), while the other only evaluates the current *best estimates* ($R_T^\mathcal{X}$). Intuitively, simultaneous minimization of both implies that not only does the algorithm get better at *estimating the optimal action* for every $\mathbf{x}$ (which may be achieved with pure exploration as well), it also does so by *making progressively better choices* (thus trading between exploration and exploitation).

Assumptions: We adopt the standard assumption that f and g have bounded norm in the reproducing kernel Hilbert space (RKHS) of functions $\mathcal{D} \to \mathbb{R}$, with positive semi-definite kernel functions $k_f, k_g : \mathcal{D} \times \mathcal{D} \to \mathbb{R}$ respectively, where $\mathcal{D} = \mathcal{D}_\mathcal{S} \times \mathcal{D}_\mathcal{X}$. We denote the RKHS by $\mathcal{H}_{k_f}(\mathcal{D})$, and its inner product by $\langle f, k_f((s, \mathbf{x}), \cdot) \rangle_{k_f}$, and the RKHS norm by $||f||_{k_f} = \sqrt{\langle f, f \rangle_{k_f}}$.

To capture the smoothness of f, we assume a known upper bound B_f on the RKHS norm of the unknown target function, i.e., $||f||_{k_f} \leq B_f$. Similarly, we assume a known upper bound B_g for the safety function g, i.e., $||g||_{k_g} \leq B_g$. We also adopt the standard assumption of bounded variance: $k_f((s, \mathbf{x}), (s, \mathbf{x})), k_g((s, \mathbf{x}), (s, \mathbf{x})) \leq 1 \; \forall (s, \mathbf{x}) \in \mathcal{D}$.

In addition, similar to Losalka and Scarlett (2023), we make the following assumptions regarding the function domain, monotonicity, and safety:

1. $\mathcal{D}_\mathcal{S} = [0, 1]$ is continuous, while $\mathcal{D}_\mathcal{X}$ can be either discrete or continuous;
2. the function g is monotonically increasing in the

[1] Several existing works instead require $g(s_t, \mathbf{x}_t) \geq h$; but this is inconsequential because $g(\cdot)$ and h can both simply be replaced by their negations.

first argument, i.e., for all $\mathbf{x} \in \mathcal{D}_\mathcal{X}$, $g(s, \mathbf{x})$ is an increasing function of $s \in \mathcal{D}_\mathcal{S}$;

3. the action $(0, \mathbf{x})$ is safe for every $\mathbf{x}$ in the domain, i.e., for all $\mathbf{x} \in \mathcal{D}_\mathcal{X}$, $g(0, \mathbf{x}) \leq h$.

Our algorithms can easily be adapted to avoid the third assumption, as long as we have access to an initial safe seed set. However, assuming $s = 0$ to be safe allows us to measure regret with respect to the *global* safe maximizer, rather than restricting to a "reachable" set. We believe this assumption is natural, as $s = 0$ is the most cautious choice for any $\mathbf{x}$.

To derive meaningful regret bounds in our setting, it turns out to be useful to impose bounds on the *maximum* growth of $f(\cdot, \mathbf{x})$ for fixed $\mathbf{x}$, as well as the *minimum* growth of $g(\cdot, \mathbf{x})$ for fixed $\mathbf{x}$. (In Appendix C.1, we argue that such requirements cannot be avoided in general.) Accordingly, we define $L_f > 0$ and $L'_g > 0$ to be the corresponding bounds on the growth rates, so that $\forall \mathbf{x} \in \mathcal{D}_\mathcal{X}, \forall s' < s$,

$$f(s, \mathbf{x}) - f(s', \mathbf{x}) \leq L_f |s - s'|, \text{ and} \tag{4}$$
$$g(s, \mathbf{x}) - g(s', \mathbf{x}) \geq L'_g |s - s'|. \tag{5}$$

We will consider algorithms that know L_f and L'_g; trivially, any upper bound on the former or lower bound on the latter also remains valid. The existence of L_f is a milder assumption than having a global Lipschitz constant, because it only concerns the growth with respect to s (known global Lipschitz constants are common in existing algorithms such as SafeOpt, SafeOpt-MC and StageOpt).

Lastly, we make the standard assumption that the noise sequence $\{\epsilon_t^f\}_{t \geq 1}$ is conditionally R_f-sub-Gaussian for a fixed constant $R_f \geq 0$, i.e.,

$$\forall t \geq 0, \forall \lambda_f \in \mathbb{R}, \mathbb{E}\left[e^{\lambda_f \epsilon_t^f} | \mathcal{F}_{t-1}\right] \leq \exp\left(\frac{\lambda_f^2 R_f^2}{2}\right), \tag{6}$$

where $\mathcal{F}_{t-1}$ is the σ-algebra generated by the random variables $\{s_k, \mathbf{x}_k, \epsilon_k^f\}_{k=1}^{t-1}$ and $\mathbf{x}_t$ (similarly, $\{\epsilon_t^g\}_{t \geq 1}$ is R_g-sub-Gaussian).

3 PROPOSED ALGORITHMS

Gaussian Process Model: Our RKHS modeling assumption naturally lends itself to the use of Gaussian process (GP) methods. Specifically, our algorithms use the zero-mean GP models $\mathrm{GP}(0, k_f)$ and $\mathrm{GP}(0, k_g)$ for f and g, along with an associated noise variance parameters $\lambda_f, \lambda_g > 0$ (which may differ from R_f, R_g).

For both f and g, upon observing t noisy values $y_1, \ldots, y_t$, the associated posterior update equations are:

$$\mu_t(s, \mathbf{x}) = k_t(s, \mathbf{x})^T \left(K_t + \lambda I\right)^{-1} \mathbf{y}_t, \tag{7}$$
$$k_t((s, \mathbf{x}), (s', \mathbf{x}')) = k\left((s, \mathbf{x}), (s', \mathbf{x}')\right) - k_t(s, \mathbf{x})^T (K_t + \lambda I)^{-1} k_t(s', \mathbf{x}'), \tag{8}$$
$$\sigma_t^2(s, \mathbf{x}) = k_t((s, \mathbf{x}), (s, \mathbf{x})), \tag{9}$$

where $\mathbf{y}_t = [y_i]_{i \leq t}$, $k_t(\cdot) = [k((s_i, \mathbf{x}_i), \cdot)]_{i \leq t}$, $K_t = [k((s_i, \mathbf{x}_i), (s_j, \mathbf{x}_j))]_{i,j \leq t}$, $k \in \{k_f, k_g\}$. When considering f and g we suitably substitute $k \in \{k_f, k_g\}$, $\lambda \in \{\lambda_f, \lambda_g\}$, and $y_i \in \{y_i^f, y_i^g\}$ respectively.

Algorithm Design: As discussed in Section 2, we consider multiple goals for our algorithms. In this section, we first outline the general structure of our proposed algorithms, and subsequently describe each specific algorithm in further detail.

The key ideas behind our algorithms are to (i) eliminate suboptimal $\mathbf{x}$'s in every round, (ii) limit the expansion of the safe set only to the regions where a "better" action may be found, and (iii) use an acquisition function that tries to reduce uncertainty in the set of actions that could either help expand the safe set or maximize the objective function. Ideas (i) and (ii) exploit the monotonicity of g in s, and distinguish our algorithm from existing ones such as SafeOpt (which uses idea (iii) but not (i)–(ii)).

Our algorithms use confidence bounds of the following standard form:

$$\mathrm{UCB}_{t-1}^f(s, \mathbf{x}) = \mu_{t-1}^f(s, \mathbf{x}) + \beta_t^f \sigma_{t-1}^f(s, \mathbf{x}), \tag{10}$$
$$\mathrm{LCB}_{t-1}^f(s, \mathbf{x}) = \mu_{t-1}^f(s, \mathbf{x}) - \beta_t^f \sigma_{t-1}^f(s, \mathbf{x}), \tag{11}$$

where UCB and LCB denote the upper and lower confidence bound respectively, β_t^f is a time-dependent constant (and analogously with g in place of f). We keep β_t^f (and β_t^g) generic here, but consider the UCB and LCB providing high-probability upper and lower bounds on f (and g). See Section 4 for specific choices.

In each round $t = 1, \ldots, T$, the main steps of our algorithms are as follows.

1. Determine the set of actions S_t that can currently be classified as safe with high probability:

$$S_t = \left\{(s, \mathbf{x}) \in \mathcal{D} : \mathrm{UCB}_{t-1}^g(s, \mathbf{x}) \leq h\right\} \cup \left\{(0, \mathbf{x}) : \mathbf{x} \in \mathcal{D}_\mathcal{X}\right\}. \tag{12}$$

2. Reduce the domain of $\mathbf{x}$'s under consideration by eliminating any $\mathbf{x} \in \mathcal{D}_\mathcal{X}^{t-1}$ that satisfies an elimination criteria, $\texttt{elim}_t(\mathbf{x}) = \texttt{true}$ (detailed below) to form the set $\mathcal{D}_\mathcal{X}^t$ (starting with $\mathcal{D}_\mathcal{X}^0 = \mathcal{D}_\mathcal{X}$):

$$\mathcal{D}_\mathcal{X}^t = \mathcal{D}_\mathcal{X}^{t-1} \setminus \left\{\mathbf{x} \in \mathcal{D}_\mathcal{X}^{t-1} : \texttt{elim}_t(\mathbf{x}) = \texttt{true}\right\}. \tag{13}$$

3. For every $\mathbf{x} \in \mathcal{D}_{\mathcal{X}}^t$, find the s-values on the current "safe boundary", given by the highest $s \in \mathcal{D}_{\mathcal{S}}$ such that safety is guaranteed with high probability (denoted by $s_t^{(\mathbf{x})}$). Form the set G_t with these $(s_t^{(\mathbf{x})}, \mathbf{x})$ pairs if there is a possibility of expansion to a more optimal f-value (as decided by a function $\texttt{expd}_t(s, \mathbf{x})$). Mathematically, we have:

$$s_t^{(\mathbf{x})} = \max\big\{s \in \mathcal{D}_{\mathcal{S}} : (s, \mathbf{x}) \in S_t\big\}, \tag{14}$$

$$G_t = \big\{(s_t^{(\mathbf{x})}, \mathbf{x}) : \mathbf{x} \in \mathcal{D}_{\mathcal{X}}^t, \texttt{expd}_t(s, \mathbf{x}) = \texttt{true}\big\}. \tag{15}$$

4. For every $\mathbf{x} \in \mathcal{D}_{\mathcal{X}}^t$, find a safe action that maximizes $\mathrm{UCB}_{t-1}^f(s, \mathbf{x})$, and form the set of such maximizers, M_t, as follows:

$$\hat{s}_t^{(\mathbf{x})} = \operatorname*{arg\,max}_{s \in \mathcal{D}_{\mathcal{S}} : s \le s_t^{(\mathbf{x})}} \mathrm{UCB}_{t-1}^f(s, \mathbf{x}), \tag{16}$$

$$M_t = \big\{(\hat{s}_t^{(\mathbf{x})}, \mathbf{x}) : \mathbf{x} \in \mathcal{D}_{\mathcal{X}}^t\big\}. \tag{17}$$

Note that we use the same notation as in (3) here (i.e., $\hat{s}_t^{(\mathbf{x})}$), because our proposed algorithms use the maximizer (over s) of $\mathrm{UCB}_{t-1}^f(s, \mathbf{x})$ as the current estimate of the optimal safe s for a given $\mathbf{x}$.

5. Use an acquisition function $\texttt{acq}_t(s, \mathbf{x})$ to select an action as follows:

$$(s_t, \mathbf{x}_t) = \operatorname*{arg\,max}_{(s, \mathbf{x})} \texttt{acq}_t(s, \mathbf{x}). \tag{18}$$

Next, we describe the three key functions, $\texttt{elim}_t, \texttt{expd}_t$ and $\texttt{acq}_t$, which vary corresponding to the different problem settings as discussed earlier.

Case 1: Maximizing f across $\mathcal{D}$: We first consider the most standard cumulative regret notion, corresponding to R_T in (1). In this case, the three functions in Algorithm 1 are defined as follows:

- $\texttt{elim}_t$: We eliminate $\mathbf{x}$'s that are suboptimal according to the confidence bounds. For defining suboptimality, we first define the following term:

$$\underline{s}_t^{(\mathbf{x})} = \max\big\{s \in \mathcal{D}_{\mathcal{S}} : \mathrm{LCB}_{t-1}^g(s_t^{(\mathbf{x})}, \mathbf{x}) + L_g' |s - s_t^{(\mathbf{x})}| \le h\big\}. \tag{19}$$

$\underline{s}_t^{(\mathbf{x})}$ essentially indicates the highest $s \in \mathcal{D}_{\mathcal{S}}$ for which the safety function could be at most h optimistically (since even with the minimum growth rate, g exceeds h beyond $\underline{s}_t^{(\mathbf{x})}$).

Suboptimality of $\mathbf{x}$ is decided based on the following two conditions: (i) the highest UCB_{t-1}^f for the $(s, \mathbf{x})$'s currently known to be safe is lower than the LCB_{t-1}^f-value of some $(s', \mathbf{x}')$ known to be safe, *and* (ii) the maximum possible f-value at $(\underline{s}_t^{(\mathbf{x})}, \mathbf{x})$ is less than the LCB_{t-1}^f-value of some $(s', \mathbf{x}')$ known to be safe. Thus, $\texttt{elim}_t(\mathbf{x})$ is $\texttt{true}$ if both of the following conditions hold:

$$\text{(i)} \max_{s \le s_t^{(\mathbf{x})}} \{\mathrm{UCB}_{t-1}^f(s, \mathbf{x})\} < \max_{(s', \mathbf{x}') \in S_t} \big\{\mathrm{LCB}_{t-1}^f(s', \mathbf{x}')\big\}, \tag{20}$$

$$\text{(ii)} \ \mathrm{UCB}_{t-1}^f(s_t^{(\mathbf{x})}, \mathbf{x}) + L_f |\underline{s}_t^{(\mathbf{x})} - s_t^{(\mathbf{x})}| \le \max_{(s', \mathbf{x}') \in S_t} \big\{\mathrm{LCB}_{t-1}^f(s', \mathbf{x}')\big\}. \tag{21}$$

These criteria ensure that even when being optimistic, any safe action corresponding to $\mathbf{x}$ (either within the current safe region as in (20), or after expanding further as in (21)) cannot result in a higher f-value than $f(s', \mathbf{x}')$.

- $\texttt{expd}_t$: We include an action $(s_t^{(\mathbf{x})}, \mathbf{x})$ in the set G_t only if expanding to $\underline{s}_t^{(\mathbf{x})}$ could optimistically lead to a better f-value than the one currently found. Thus, $\texttt{expd}_t(\mathbf{x})$ is set to $\texttt{true}$ if the follow-

Algorithm 1 M-SafeOpt

Input: Prior $\mathrm{GP}(0, k_f)$, $\mathrm{GP}(0, k_g)$, parameters $\lambda_f, \lambda_g, L_f, L_g', \{\beta_t^f\}_{t \ge 1}, \{\beta_t^g\}_{t \ge 1}$
$\mathcal{D}_{\mathcal{X}}^0 = \mathcal{D}_{\mathcal{X}}$
for $t = 1, \dots, T$ **do**
  $\mathcal{D}_{\mathcal{X}}^t = \mathcal{D}_{\mathcal{X}}^{t-1} \setminus \{\mathbf{x} \in \mathcal{D}_{\mathcal{X}}^{t-1} : \texttt{elim}_t(\mathbf{x}) = \texttt{true}\}$
    ▷ eliminate all suboptimal $\mathbf{x}$ to form $\mathcal{D}_{\mathcal{X}}^t$
  $G_t = \emptyset$
  $M_t = \emptyset$
  for $\mathbf{x} \in \mathcal{D}_{\mathcal{X}}^t$ **do** ▷ find max. safe $s \, \forall \mathbf{x} \in \mathcal{D}_{\mathcal{X}}^t$
    if $\mathrm{UCB}_{t-1}^g(s, \mathbf{x}) > h \, \forall s \in \mathcal{D}_{\mathcal{S}}$ **then**
      $s_t^{(\mathbf{x})} = 0$
    else if $\exists s \in \mathcal{D}_{\mathcal{S}} : \mathrm{UCB}_{t-1}^g(s, \mathbf{x}) = h$ **then**
      $s_t^{(\mathbf{x})} = \max\{s \in \mathcal{D}_{\mathcal{S}}, \mathrm{UCB}_{t-1}^g(s, \mathbf{x}) = h\}$
    else
      $s_t^{(\mathbf{x})} = 1$
    end if
    if $\texttt{expd}_t(s_t^{(\mathbf{x})}, \mathbf{x})$ is $\texttt{true}$ **then** ▷ form G_t
      $G_t = G_t \cup \{(s_t^{(\mathbf{x})}, \mathbf{x})\}$
    end if
    $\hat{s}_t^{(\mathbf{x})} = \arg\max_{s \in \mathcal{D}_{\mathcal{S}} : s \le s_t^{(\mathbf{x})}} \mathrm{UCB}_{t-1}^f(s, \mathbf{x})$
    $M_t = M_t \cup \{(\hat{s}_t^{(\mathbf{x})}, \mathbf{x})\}$ ▷ form M_t
  end for
  $(s_t, \mathbf{x}_t) = \arg\max_{(s, \mathbf{x}) : \mathbf{x} \in \mathcal{D}_{\mathcal{X}}^t} \texttt{acq}_t(s, \mathbf{x})$
  Update posterior to get $\mu_t^f, \sigma_t^f, \mu_t^g, \sigma_t^g$
end for

ing condition holds:

$$\mathrm{UCB}^f_{t-1}(s_t^{(\mathbf{x})}, \mathbf{x}) + L_f|\underline{s}_t^{(\mathbf{x})} - s_t^{(\mathbf{x})}| > \max_{(s',\mathbf{x}')\in S_t} \left\{\mathrm{LCB}^f_{t-1}(s', \mathbf{x}')\right\}. \quad (22)$$

- $\texttt{acq}_t$: We define the acquisition function as:

$$\texttt{acq}_t(s, \mathbf{x}) = \begin{cases} \max\left\{\beta_t^f \sigma^f_{t-1}(s, \mathbf{x}), \beta_t^g \sigma^g_{t-1}(s, \mathbf{x})\right\}, & \text{if } (s, \mathbf{x}) \in G_t; \\ \beta_t^f \sigma^f_{t-1}(s, \mathbf{x}), \text{ if } (s, \mathbf{x}) \in M_t \text{ and } (s, \mathbf{x}) \notin G_t; \\ 0, \text{ otherwise,} \end{cases} \quad (23)$$

in order to reduce uncertainty of f within the potential maximizers M_t, and reduce the uncertainty of *both* f *and* g in the expander set G_t.

Case 2: Maximizing f for every $\mathbf{x} \in \mathcal{D}_\mathcal{X}$: Suppose that the goal is to find the optimal safe s for *every* $\mathbf{x}$ (goal (ii) in Section 2). In this case, we modify $\texttt{elim}_t$ and $\texttt{expd}_t$ as follows, while keeping $\texttt{acq}_t$ unchanged.

- $\texttt{elim}_t$: Since we want to find the optimal action corresponding to *every* $\mathbf{x}$, we should not eliminate any $\mathbf{x}$'s from consideration in this case. Thus, we define $\texttt{elim}_t(\mathbf{x}) = \texttt{false}\ \forall \mathbf{x} \in \mathcal{D}_\mathcal{X}$.
- $\texttt{expd}_t$: With a similar motivation to Case 1, $\texttt{expd}_t(\mathbf{x})$ is set to $\texttt{true}$ if the following holds:

$$\mathrm{UCB}^f_{t-1}(s_t^{(\mathbf{x})}, \mathbf{x}) + L_f|\underline{s}_t^{(\mathbf{x})} - s_t^{(\mathbf{x})}| > \max_{s \le s_t^{(\mathbf{x})}} \{\mathrm{LCB}^f_{t-1}(s, \mathbf{x})\}, \quad (24)$$

where $\underline{s}_t^{(\mathbf{x})}$ is the highest "potentially safe" s as earlier (see (19)). The condition (24) states that the optimistic f-value at $\underline{s}_t^{(\mathbf{x})}$ is better than a pessimistic value among the s known to be safe given $\mathbf{x}$. Note that unlike Case 1, we use $\mathbf{x}$ on both sides and do *not* compare to any $\mathbf{x}' \neq \mathbf{x}$; this is because here we must find the best s for *every* $\mathbf{x}$.

Case 3: Both f and g are monotone: Here we consider the scenario where both f and g are monotonically increasing functions in s. As a first sub-case, we again consider the goal (1) associated with finding a global safe maximizer. While the same algorithm as case 1 would work here as well, we find that the algorithm can be simplified substantially. Since we know that the optimal f-value will be found along the safe boundary, there is no requirement to separately maintain the set M_t.

We simplify the three key functions in Algorithm 1 as follows. First, for defining $\texttt{elim}_t$, it suffices to consider (21) only. Next, we always expand for any $\mathbf{x}$ that is not eliminated, i.e., $\forall \mathbf{x} \in \mathcal{D}^t_\mathcal{X}$, $\texttt{expd}_t$ is set to $\texttt{true}$. Finally, the acquisition function is simplified as follows:

$$\texttt{acq}_t(s, \mathbf{x}) = \begin{cases} \max\left\{\beta_t^f \sigma^f_{t-1}(s, \mathbf{x}), \beta_t^g \sigma^g_{t-1}(s, \mathbf{x})\right\}, & \text{if } (s, \mathbf{x}) \in G_t; \\ 0, \text{ otherwise.} \end{cases} \quad (25)$$

As a second sub-case, with *both* f and g still being monotone, we may consider the task of finding the best (highest) s for every $\mathbf{x}$. In this scenario, removing the elimination step altogether, setting $\texttt{expd}_t(\mathbf{x}) = \texttt{false}$ when $s_t^{(\mathbf{x})} = 1$, and modifying the acquisition function to only consider σ^g_{t-1} essentially results in the M-SafeUCB algorithm of Losalka and Scarlett (2023). In Appendix A.6, we show how their theoretical guarantees can be recovered as a special case of ours.

Extensions: In Appendix C.2, we outline several straightforward extensions of our algorithm, including multiple safety functions, joint RKHS modeling of (f, g), and the presence of contextual variables.

4 THEORETICAL RESULTS

Our theoretical analysis relies on the widely-used notion of *information gain*, which we define separately for f and g as follows:

$$\gamma_t^f := \max_{A \subset \mathcal{D}: |A| = t} I(y_A^f; f_A), \quad (26)$$

$$\gamma_t^g := \max_{A \subset \mathcal{D}: |A| = t} I(y_A^g; g_A), \quad (27)$$

where $I(y_A^f; f_A)$ denotes the mutual information between $f_A = [f(s, \mathbf{x})]_{(s,\mathbf{x})\in A}$ and $y_A^f = f_A + \epsilon_A^f$, and where $\epsilon_A^f \sim \mathcal{N}(0, \lambda_f I)$ (similarly, for $I(y_A^g; g_A)$). The information gain essentially captures the amount of uncertainty reduction in the function as a result of observing t noisy evaluations.

Recall that our confidence bounds (10)–(11) depend on parameters β_t^f and β_t^g that have been generic until now. We will state our results keeping them generic, but requiring their validity; formally, we say that (β_t^f, β_t^g) provide $(1-\delta)$-*valid confidence bounds* if, with probability at least $1 - \delta$, for all $(s, \mathbf{x}) \in \mathcal{D}, t \ge 1$,

$$|\mu^f_{t-1}(s, \mathbf{x}) - f(s, \mathbf{x})| \le \beta_t \sigma^f_{t-1}(s, \mathbf{x}), \text{ and} \quad (28)$$

$$|\mu^g_{t-1}(s, \mathbf{x}) - g(s, \mathbf{x})| \le \beta_t \sigma^g_{t-1}(s, \mathbf{x}). \quad (29)$$

Additionally, our proofs rely on the β_t-terms being non-decreasing. The following lemma from Chowdhury and Gopalan (2017) (namely, their Theorem 2 and a union bound over f and g) provides a well-known such choice of (β_t^f, β_t^g); we will also mention an alternative below.

Lemma 1. *For any $\delta > 0$, the parameters*

$$\beta_t^f = B_f + R_f\sqrt{2(\gamma_{t-1}^f + 1 + \ln(2/\delta))}, \text{ and} \quad (30)$$

$$\beta_t^g = B_g + R_g\sqrt{2(\gamma_{t-1}^g + 1 + \ln(2/\delta))} \quad (31)$$

provide $(1-\delta)$-valid confidence bounds.

We are now ready to state our main theorems, all of which are proved in Appendix A. Recall that our three regret notions R_T, R'_T, and $R_T^{\mathcal{X}}$ are defined in (1)–(3), and L_f, L'_g are defined in (4)–(5).

Theorem 1. *Under the setup and assumptions of Section 2 and any non-decreasing β_t^f, β_t^g providing $(1-\delta)$-valid confidence bounds, Algorithm 1 (in both case 1 and case 3) satisfies the following with probability at least $1-\delta$:*

$$R_T = O\left(\left(1 + \frac{L_f}{L'_g}\right)\left(\beta_T^g\sqrt{T\gamma_T^g} + \beta_T^f\sqrt{T\gamma_T^f}\right)\right). \quad (32)$$

Theorem 2. *Under the setup and assumptions of Section 2 and any non-decreasing β_t^f, β_t^g providing $(1-\delta)$-valid confidence bounds, Algorithm 1 (in case 2) satisfies the following with probability at least $1-\delta$:*

$$R'_T = O\left(\left(1 + \frac{L_f}{L'_g}\right)\left(\beta_T^g\sqrt{T\gamma_T^g} + \beta_T^f\sqrt{T\gamma_T^f}\right)\right). \quad (33)$$

Theorem 3. *Under the setup and assumptions of Section 2, and any non-decreasing β_t^f, β_t^g providing $(1-\delta)$-valid confidence bounds, Algorithm 1 (in case 2) satisfies the following with probability at least $1-\delta$:*

$$R_T^{\mathcal{X}} = O\left(\left(1 + \frac{L_f}{L'_g}\right)\left(\beta_T^g\sqrt{T\gamma_T^g} + \beta_T^f\sqrt{T\gamma_T^f}\right)\right). \quad (34)$$

We note that the above regret bounds can be refined to the following form:

$$R_T = O\left(\beta_T^f\sqrt{T\gamma_T^f} + \frac{L_f}{L'_g}\beta_T^g\sqrt{T\gamma_T^g}\right) \quad (35)$$

by modifying the acquisition function as follows:

$$\mathtt{acq}_t(s,\mathbf{x}) = \max\{\beta_t^f\sigma_{t-1}^f(s,\mathbf{x}), (L_f/L'_g)\beta_t^g\sigma_{t-1}^g(s,\mathbf{x})\} \quad (36)$$

if $(s,\mathbf{x}) \in G_t$ (with the other cases remaining unchanged in (23) and (25)). This refined bound applies to R_T (in both case 1 and case 3), R'_T (in case 2), and $R_T^{\mathcal{X}}$ (in case 2), and provides desirable properties with respect to the scaling of f and g. (See Appendix A.5 for a proof of the validity of the refined regret bounds, and a discussion on the scaling properties.) However, this comes with the trade-off of having to incorporate the constants L_f and L'_g into the acquisition function. This introduces additional dependence of the algorithm on these constants, which may be undesirable. Therefore, we we primarily focus on the algorithm with the earlier acquisition function (18), which has similar sublinear regret guarantees.

For all of the above theorems, our regret bounds have the same $\beta_T\sqrt{T\gamma_T}$ form (with f or g superscripts) as GP-UCB (Srinivas et al., 2012) and other related algorithms (e.g., (Chowdhury and Gopalan, 2017)), and hence also the same bounds when applied to specific kernels. For instance, for the squared exponential kernel, we have $\gamma_T = O(\ln^{d+1} T)$ (Srinivas et al., 2012), which implies sublinear regret via Lemma 1.

For the Matérn-ν kernel, we have $\gamma_T = O\big(T^{\frac{d}{2\nu+d}}\log T\big)$ (Vakili et al., 2021). Since β_T is also linear in γ_T in Lemma 1, this only guarantees sublinear regret if $\frac{d}{2\nu+d} < \frac{1}{2}$, i.e., $\nu > \frac{d}{2}$. Fortunately, alternative confidence bounds have recently been given that guarantee sublinear regret without this restriction (Whitehouse et al., 2023), and we can directly make use of these since our theorems are stated in terms of generic confidence bounds. The changes required for these variations are identical to the vanilla setting without safety constraints, so we do not repeat them.

5 EXPERIMENTS

In this section, we present empirical results to complement our theoretical findings, by running the M-SafeOpt algorithm (in Case 1 and Case 2) and comparing with baseline algorithms [2]. The primary goal of the experiments is to (i) verify that our cumulative regret notions demonstrate sublinear behavior, (ii) verify that unsafe actions are not sampled, and (iii) demonstrate performance gains over existing algorithms. We primarily use SafeOpt-MC (Berkenkamp et al., 2021) and the purely exploratory PredVar (Schreiter et al., 2015) algorithms for comparisons. SafeOpt (Sui et al., 2015) and M-SafeUCB (Losalka and Scarlett, 2023) are skipped, since they are designed to work with a single function f. Additional experimental results are provided in Appendix B.1, and the details (e.g., descriptions of baselines, choices of kernels and β_t) are given in Appendix B.2.

Simulated Clinical Trial: For this experiment, we use the logistic function as a model of the dose-toxicity and dose-efficacy behaviors. Specifically, following Cai

[2]The code is available at `https://github.com/arpanlosalka/m-safeopt`.

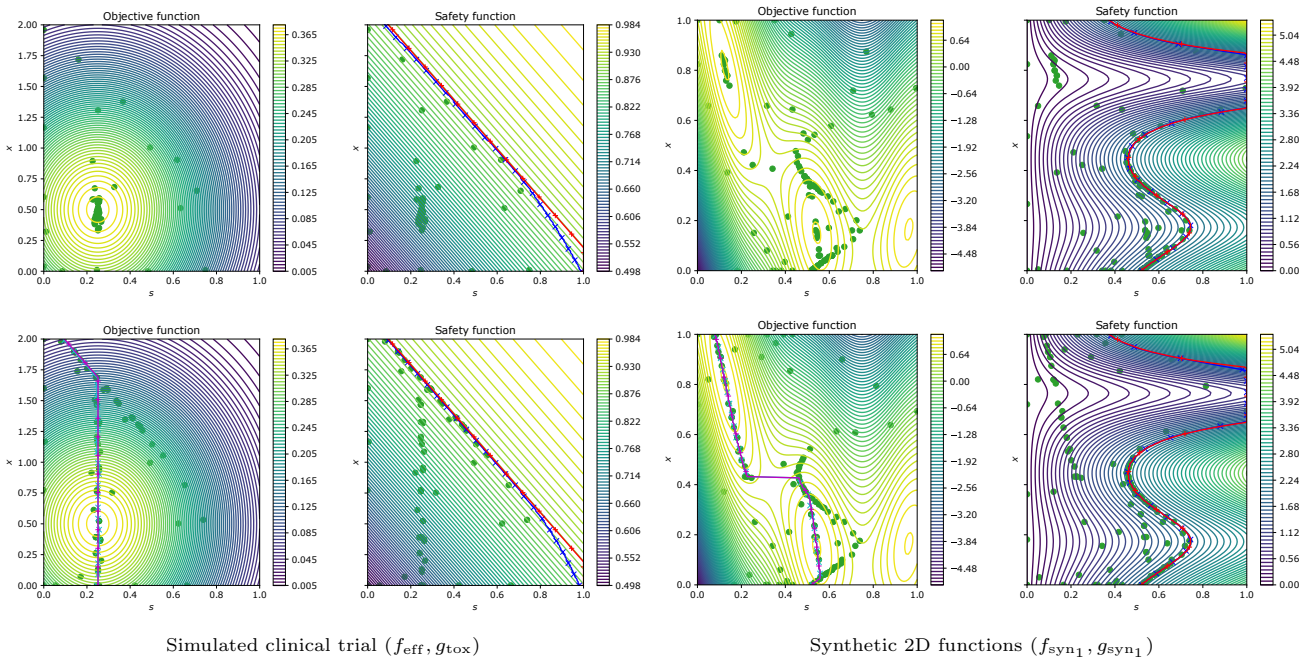

Figure 1: Actions sampled by M-SafeOpt^3 in case 1 (top row) and case 2 (bottom row), along with the safe boundaries discovered in blue and true safe boundaries in red (2$^{\text{nd}}$ and 4$^{\text{th}}$ column). In case 2, the 1$^{\text{st}}$ and 3$^{\text{rd}}$ columns also show the optimal s discovered for every $\mathbf{x}$ in cyan, and the true optimal s-values in magenta.

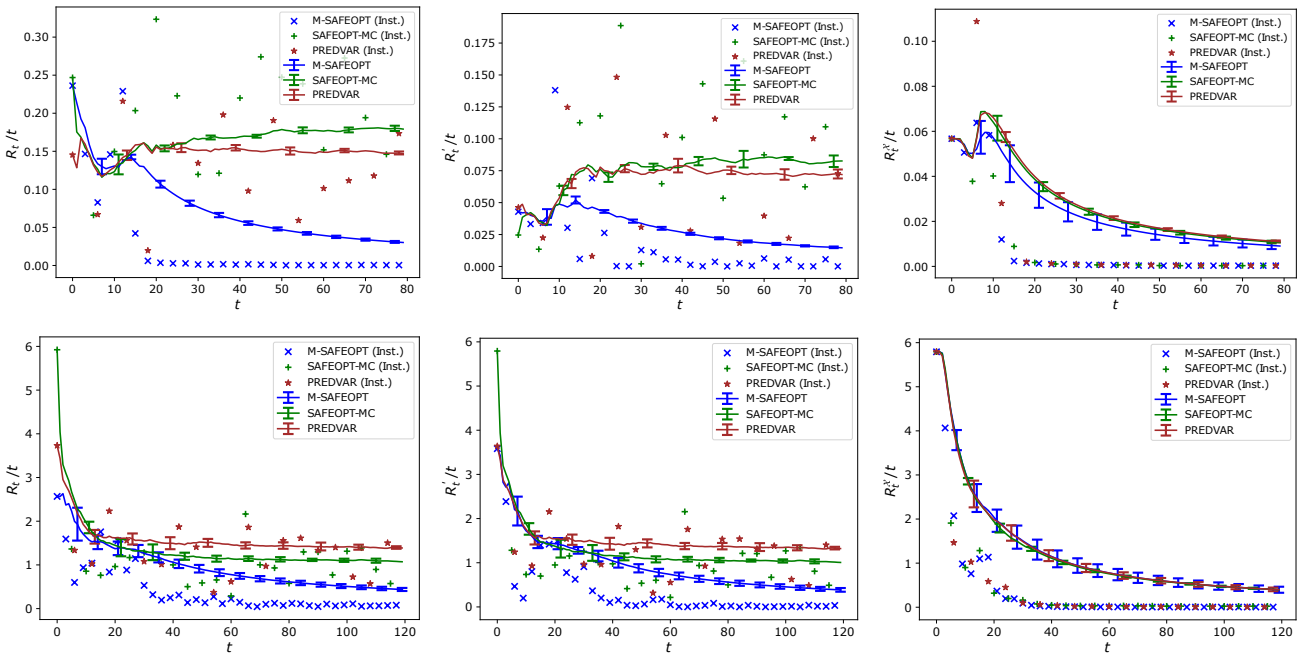

Figure 2: The top row shows the regret plots for the simulated clinical trial experiment, and the bottom row shows that for the synthetic 2D experiment for M-SafeOpt, along with baseline algorithms. The first column shows the plot for R_t/t (for Case 1), while columns 2 and 3 show R'_t/t and $R^{\mathcal{X}}_t/t$ (for Case 2). The corresponding instantaneous regret values are shown using markers.

et al. (2014), we use the functions

$$f_{\text{eff}}(d_1, d_2) = \left\{1 + e^{\theta_0^f - \theta_1^f d_1 - \theta_2^f d_2 - \theta_3^f d_1^2 - \theta_4^f d_2^2}\right\}^{-1} \quad (37)$$

$$\text{and} \quad g_{\text{tox}}(d_1, d_2) = \left\{1 + e^{-\theta_1^g d_1 - \theta_2^g d_2}\right\}^{-1}, \quad (38)$$

where $d_1(s)$ and $d_2(\mathbf{x})$ denote the dosage of two drugs, and θ_i's denote suitable parameters (see Appendix B.2 for details). While g_{tox} increases monotonically with the dosage of both drugs, the efficacy peaks at an intermediate dose level of the drugs and then decreases.

Synthetic 2D function: In this section, we use the scaled Branin function from (Picheny et al., 2013) as the objective function f_{syn_1}. It has a more complex optimization surface, with three local optima over the domain considered, $\mathcal{D} = [0,1]^2$. For the safety function g_{syn_1}, we use a slightly modified form of f_{syn_2} from (Losalka and Scarlett, 2023), such that optimization becomes more challenging for both goal (i) (global optimization) and goal (ii) (optimization $\forall \mathbf{x}$).

Observations: For both the above experiments, we observe in Figure 1 that unsafe actions are not sampled, and for both goal (i) and goal (ii), the samples of M-SafeOpt tend to be near the optimal actions instead of unnecessarily exploring suboptimal regions of the input space.

The regret plots in Figure 2 demonstrate that M-SafeOpt achieves sublinear regret (for suitable notions of regret, depending on the goal), whereas the baseline algorithms fail to do so (except for $R_T^{\mathcal{X}}$). Both of the baseline algorithms continue to explore suboptimal regions, either near the safe boundary (SafeOpt-MC) or throughout the safe region (PredVar), and this gets reflected in the R_t/t and R'_t/t plots. However, they perform well with respect to $R_t^{\mathcal{X}}$, which considers only the *best action discovered*, and not the *action chosen* in each round for evaluation. As discussed in Section 2, our goal (ii) naturally leads to a requirement of both R'_T and $R_T^{\mathcal{X}}$ being small, rather than either of them alone.

Next, we highlight some observations regarding the performance of M-SafeOpt based on our experiments. (See Appendix B.1.1 for further discussion on SafeOpt-MC and PredVar.)

- M-SafeOpt (Case 1), when used for goal (i) (as in Section 2) is able to eliminate suboptimal $\mathbf{x}$'s (via $\texttt{elim}_t$) and converge quickly to the regions in which there is a higher probability of finding the safe optimal action. For instance, this can be observed for $\mathbf{x} \in [1,2]$ in row 1-column 1 of Figure 1 for $(f_{\text{eff}}, g_{\text{tox}})$ and for $\mathbf{x} \in [0.9,1]$ in row 1-column 3 for $(f_{\text{syn}_1}, g_{\text{syn}_1})$, where the algorithm stops exploring once it has located better actions near/at the true optimum.
- M-SafeOpt (in Case 1 and Case 2) also limits unnecessary expansion beyond the currently discovered safe boundary (via $\texttt{expd}_t$), helping the algorithm to converge to the optimal region quicker that SafeOpt . For instance, this is observed for $\mathbf{x} \in [0, 0.25]$ in row 1-column 2 (Case 1) of Figure 1 for $(f_{\text{eff}}, g_{\text{tox}})$, and similarly in row 2-column 2 of the same figure (Case 2).
- The safe boundary eventually discovered by M-SafeOpt (in both Case 1 and Case 2) tends to diverge from the true safe boundary with respect to g. This is also due to elimination and limiting expansion. Note that when no $\mathbf{x}$ can be eliminated (in Case 2), the same phenomenon (e.g., $\mathbf{x} \in [0, 0.25]$ in row 2-column 2 of Figure 1) can be observed due to non-expansion of suboptimal $\mathbf{x}$'s.
- In Case 2 (for goal (ii)), M-SafeOpt is able to find the optimal s for every $\mathbf{x} \in \mathcal{D}_{\mathcal{X}}$ almost exactly (as seen in Figure 1, and also reflected in the regret plot in the third column of Figure 2 with respect to $R_t^{\mathcal{X}}$), while making progressively better choices (as evidenced by the performance with respect to R'_t in column 2 in Figure 2).

6 CONCLUSION

In this work, we have shown how monotonicity of the unknown safety function in a single safety variable allows us to develop a no-regret safe black-box optimization algorithm. We provided several variations of our algorithm that work with different goals of practical relevance. Like many GP-based algorithms, our techniques are primarily suited to low dimensions, and variations suited to higher dimensions would be of interest. Other potential areas of future work include exploring other helpful structures of g beyond monotonicity, and studying extensions to more general settings such as reinforcement learning.

Acknowledgement

This work was supported by the Singapore Ministry of Education Academic Research Fund Tier 1 under grant number A-8000872-00-00.

[3] See Appendix B.1 for similar plots with the baselines.

References

Daniel J Lizotte, Tao Wang, Michael H Bowling, Dale Schuurmans, et al. Automatic gait optimization with Gaussian process regression. In *International Joint Conference on Artificial Intelligence*, volume 7, pages 944–949, 2007.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias W Seeger. Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

Ami Takahashi and Taiji Suzuki. Bayesian optimization for estimating the maximum tolerated dose in Phase I clinical trials. *Contemporary Clinical Trials Communications*, 21:100753, 2021.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, 25, 2012.

Hastagiri P Vanchinathan, Isidor Nikolic, Fabio De Bona, and Andreas Krause. Explore-exploit in top-n recommender systems via Gaussian processes. In *ACM Conference on Recommender Systems*, pages 225–232, 2014.

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

Yanan Sui, Vincent Zhuang, Joel Burdick, and Yisong Yue. Stagewise safe Bayesian optimization with Gaussian processes. In *International Conference on Machine Learning*, pages 4781–4789. PMLR, 2018.

Felix Berkenkamp, Andreas Krause, and Angela P Schoellig. Bayesian optimization with safety constraints: Safe and automatic parameter tuning in robotics. *Machine Learning*, pages 1–35, 2021.

Donald A Berry. Adaptive clinical trials in oncology. *Nature reviews Clinical oncology*, 9(4):199–207, 2012.

S. Chevret. *Statistical Methods for Dose-Finding Experiments*. Statistics in Practice. Wiley, 2006.

Chunyan Cai, Ying Yuan, and Yuan Ji. A Bayesian dose finding design for oncology clinical trials of combinational biological agents. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 63(1):159–173, 2014.

Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. *Advances in Neural Information Processing Systems*, 30, 2017.

Matteo Turchetta, Felix Berkenkamp, and Andreas Krause. Safe exploration for interactive machine learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Matteo Turchetta, Felix Berkenkamp, and Andreas Krause. Safe exploration in finite Markov decision processes with Gaussian processes. *Advances in Neural Information Processing Systems*, 29, 2016.

Matteo Turchetta, Andrey Kolobov, Shital Shah, Andreas Krause, and Alekh Agarwal. Safe reinforcement learning via curriculum induction. *Advances in Neural Information Processing Systems*, 33:12151–12162, 2020.

Sanae Amani, Mahnoosh Alizadeh, and Christos Thrampoulidis. Regret bounds for safe Gaussian process bandit optimization. In *IEEE International Symposium on Information Theory (ISIT)*, pages 527–532, 2021.

Dominik Baumann, Alonso Marco, Matteo Turchetta, and Sebastian Trimpe. GoSafe: Globally optimal safe robot learning. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4452–4458, 2021.

Bhavya Sukhija, Matteo Turchetta, David Lindner, Andreas Krause, Sebastian Trimpe, and Dominik Baumann. GoSafeOpt: Scalable safe exploration for global optimization of dynamical systems. *Artif. Intell.*, 320:103922, 2022.

Arpan Losalka and Jonathan Scarlett. Benefits of monotonicity in safe exploration with Gaussian processes. In *Uncertainty in Artificial Intelligence*, pages 1304–1314. PMLR, 2023.

Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning*, pages 844–853. PMLR, 2017.

Sattar Vakili, Kia Khezeli, and Victor Picheny. On information gain and regret bounds in Gaussian process bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 82–90. PMLR, 2021.

Justin Whitehouse, Zhiwei Steven Wu, and Aaditya Ramdas. On the sublinear regret of GP-UCB. *arXiv preprint arXiv:2307.07539*, 2023.

Jens Schreiter, Duy Nguyen-Tuong, Mona Eberts, Bastian Bischoff, Heiner Markert, and Marc Toussaint. Safe exploration for active learning with Gaussian processes. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 133–149. Springer, 2015.

Victor Picheny, Tobias Wagner, and David Ginsbourger. A benchmark of kriging-based infill criteria

for noisy optimization. *Structural and multidisciplinary optimization*, 48:607–626, 2013.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI gym. *arXiv preprint arXiv:1606.01540*, 2016.

Abdel-Rahman Hedar. Global optimization test problems. `http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO.htm`, 2013. Accessed: 2024-02-20.

Victor Picheny, Joel Berkeley, Henry B. Moss, Hrvoje Stojic, Uri Granta, Sebastian W. Ober, Artem Artemev, Khurram Ghani, Alexander Goodall, Andrei Paleyes, Sattar Vakili, Sergio Pascual-Diaz, Stratis Markou, Jixiang Qing, Nasrulloh R. B. S Loka, and Ivo Couckuyt. Trieste: Efficiently exploring the depths of black-box functions with tensorflow, 2023. URL `https://arxiv.org/abs/2302.08436`.

Shubhanshu Shekhar and Tara Javidi. Multi-scale zero-order optimization of smooth functions in an RKHS. *arXiv preprint arXiv:2005.04832*, 2020.

CHECKLIST

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [**Yes**]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [**Yes**]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [**Yes**]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [**Yes**]
   (b) Complete proofs of all theoretical results. [**Yes**]
   (c) Clear explanations of any assumptions. [**Yes**]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [**Yes**]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [**Yes**]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [**Yes**]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [**Not Applicable**]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator if your work uses existing assets. [**Yes**]
   (b) The license information of the assets, if applicable. [**Not Applicable**]
   (c) New assets either in the supplemental material or as a URL, if applicable. [**Not Applicable**]
   (d) Information about consent from data providers/curators. [**Not Applicable**]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [**Not Applicable**]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [**Not Applicable**]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [**Not Applicable**]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [**Not Applicable**]

Appendix

A PROOFS

In this section, we present the proofs for the three theorems.

A.1 Proof of Theorem 1 (Case 1)

Let $(s_*, \mathbf{x}_*)$ denote the optimal action (or any one such action if there are multiple). Recall the definition of $s_t^{(\mathbf{x})} = \max\left\{s \in \mathcal{D}_{\mathcal{S}} : (s, \mathbf{x}) \in S_t\right\}$ from (14).

For deriving the following results, we assume the validity of the confidence bounds, which are known to hold with probability at least $1-\delta$. That is, we condition on (28) and (29) both being true. To characterize the regret incurred at a given time instant t, we will split the analysis into two cases: (i) the optimal action $(s_*, \mathbf{x}_*) \notin S_t$, i.e., $\mathrm{UCB}_{t-1}^g(s_*, \mathbf{x}_*) > h$, and (ii) $(s_*, \mathbf{x}_*) \in S_t$, i.e., $\mathrm{UCB}_{t-1}^g(s_*, \mathbf{x}_*) \leq h$.

A.1.1 Regret for $(s_*, \mathbf{x}_*) \notin S_t$

First, we consider the regret incurred in rounds where $(s_*, \mathbf{x}_*) \notin S_t$. Since the optimal point $(s_*, \mathbf{x}_*)$ is safe by definition, we have

$$g(s_*, \mathbf{x}_*) \leq h. \tag{39}$$

Next, recall the definition of the set S_t from (12), and the definition of $s_t^{(\mathbf{x})}$ in (14). Intuitively, S_t consists of all actions that can be classified as safe via UCB_{t-1}^g and the safety threshold h, while $s_t^{(\mathbf{x})}$ corresponds to the action on the current "safe boundary" for a specific $\mathbf{x}$.

The following cases may arise: (i) $\mathrm{UCB}_{t-1}^g(s_t^{(\mathbf{x})}, \mathbf{x}) > h$ (e.g., in the initial rounds when $s_t^{(\mathbf{x})} = 0$), (ii) $\mathrm{UCB}_{t-1}^g(s_t^{(\mathbf{x})}, \mathbf{x}) = h$ (when $\mathrm{UCB}_{t-1}^g(s, \mathbf{x})$ "crosses" h for some s), or (iii) $\mathrm{UCB}_{t-1}^g(s_t^{(\mathbf{x})}, \mathbf{x}) < h$ (for $s_t^{(\mathbf{x})} = 1$ when $g(1, \mathbf{x}) < h$). In the following, we first consider the rounds for which either (i) or (ii) holds for $(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t)$, i.e, for the specific $\mathbf{x}_t$ chosen at round t. Thereafter, we show how the results that we derive also continue to remain valid when (iii) holds.

Since we are assuming that either (i) or (ii) holds for $(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t)$ and that $(s_*, \mathbf{x}_*) \notin S_t$ (for now), we have the following:

$$\mathrm{UCB}_{t-1}^g(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t) \geq h, \tag{40}$$

$$\mathrm{UCB}_{t-1}^g(s_t^{(\mathbf{x}_*)}, \mathbf{x}_*) \geq h.. \tag{41}$$

In more detail, (41) follows because the condition $(s_*, \mathbf{x}_*) \notin S_t$ states that $(s_*, \mathbf{x}_*)$ has not been discovered as safe by the algorithm, meaning it cannot be that $\mathrm{UCB}_{t-1}^g(s_t^{(\mathbf{x}_*)}, \mathbf{x}_*) < h$.

Bounding $|\underline{s}_t^{(\mathbf{x}_t)} - s_t^{(\mathbf{x}_t)}|$. By the definition of $\underline{s}_t^{(\mathbf{x})}$ in (19), the following holds:

$$\mathrm{LCB}_{t-1}^g(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t) + L_g' |\underline{s}_t^{(\mathbf{x}_t)} - s_t^{(\mathbf{x}_t)}| \leq h, \tag{42}$$

where strict inequality (i.e., $\mathrm{LCB}_{t-1}^g(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t) + L_g' |\underline{s}_t^{(\mathbf{x}_t)} - s_t^{(\mathbf{x}_t)}| < h$) may hold when $\underline{s}_t^{(\mathbf{x}_t)} = 1$.

Therefore, we deduce the following:

$$\begin{aligned}
L_g' |\underline{s}_t^{(\mathbf{x}_t)} - s_t^{(\mathbf{x}_t)}| &\leq h - \mathrm{LCB}_{t-1}^g(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t) \\
&\leq \mathrm{UCB}_{t-1}^g(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t) - \mathrm{LCB}_{t-1}^g(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t) \qquad \text{(by (40))} \\
&= 2\beta_t^g \sigma_t^g(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t) \qquad \text{(by (10) and (11))} \\
\implies |\underline{s}_t^{(\mathbf{x}_t)} - s_t^{(\mathbf{x}_t)}| &\leq 2\beta_t^g \sigma_t^g(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t)/L_g'.
\end{aligned} \tag{43}$$

This result upper bounds the distance of $\underline{s}_t^{(\mathbf{x}_t)}$ from $s_t^{(\mathbf{x}_t)}$ in terms of the width of the confidence interval at $(s_t^{(\mathbf{x}_t)}, \mathbf{x})$.

Note that (43) holds trivially if $\mathrm{UCB}_{t-1}^g(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t) < h$ (i.e., if (40) does not hold). This is because the UCB_{t-1}^g-value being less than h for an action on the current "safe boundary" implies that the algorithm has discovered $(s, \mathbf{x}_t)$ to be safe for all $s \in \mathcal{D}_{\mathcal{S}}$, and hence $s_t^{(\mathbf{x}_t)} = 1$. This would also imply that $\underline{s}_t^{(\mathbf{x})} = 1$ (following (19)), thus giving $|\underline{s}_t^{(\mathbf{x}_t)} - s_t^{(\mathbf{x}_t)}| = 0$.

Bounding $|s_* - s_t^{(\mathbf{x}_*)}|$. Similarly to the above, due to the safety of the optimal action, we have the following:

$$\begin{aligned} L_g'|s_* - s_t^{(\mathbf{x}_*)}| &\le g(s_*, \mathbf{x}_*) - g(s_t^{(\mathbf{x}_*)}, \mathbf{x}_*) \qquad \text{(by definition of } L_g') \\ &\le h - \mathrm{LCB}_{t-1}^g(s_t^{(\mathbf{x}_*)}, \mathbf{x}_*) \qquad \text{(by (29) and (39))} \\ &\le \mathrm{UCB}_{t-1}^g(s_t^{(\mathbf{x}_*)}, \mathbf{x}_*) - \mathrm{LCB}_{t-1}^g(s_t^{(\mathbf{x}_*)}, \mathbf{x}_*) \qquad \text{(by (41))} \\ &= 2\beta_t^g \sigma_{t-1}^g(s_t^{(\mathbf{x}_*)}, \mathbf{x}_*) \qquad \text{(by (10) and (11))} \\ \implies |s_* - s_t^{(\mathbf{x}_*)}| &\le 2\beta_t^g \sigma_{t-1}^g(s_t^{(\mathbf{x}_*)}, \mathbf{x}_*)/L_g'. \end{aligned} \tag{44}$$

This result upper bounds the distance of s_* from $s_t^{(\mathbf{x}_*)}$ in terms of the width of the confidence interval at $(s_t^{(\mathbf{x}_*)}, \mathbf{x}_*)$.

Bounding instantaneous regret. We consider the instantaneous regret $f(s_*, \mathbf{x}_*) - f(s_t, \mathbf{x}_t)$ in each round t, and split it as a sum of three terms, which we proceed to bound individually:

$$f(s_*, \mathbf{x}_*) - f(s_t, \mathbf{x}_t) = \Big(f(s_*, \mathbf{x}_*) - f(s_t^{(\mathbf{x}_*)}, \mathbf{x}_*)\Big) + \Big(f(s_t^{(\mathbf{x}_*)}, \mathbf{x}_*) - f(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t)\Big) + \Big(f(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t) - f(s_t, \mathbf{x}_t)\Big). \tag{45}$$

In some cases, we will also use a near-identical decomposition with $\hat{s}_t^{(\mathbf{x}_t)}$ (from (16)) replacing $s_t^{(\mathbf{x}_t)}$ (depending on the criteria for $\mathbf{x}_t$ not being eliminated at round t) as follows:

$$f(s_*, \mathbf{x}_*) - f(s_t, \mathbf{x}_t) = \Big(f(s_*, \mathbf{x}_*) - f(s_t^{(\mathbf{x}_*)}, \mathbf{x}_*)\Big) + \Big(f(s_t^{(\mathbf{x}_*)}, \mathbf{x}_*) - f(\hat{s}_t^{(\mathbf{x}_t)}, \mathbf{x}_t)\Big) + \Big(f(\hat{s}_t^{(\mathbf{x}_t)}, \mathbf{x}_t) - f(s_t, \mathbf{x}_t)\Big). \tag{46}$$

We also note that given the acquisition function in (23), $(s_t, \mathbf{x}_t)$ being chosen at round t implies that

$$\max\Big\{\beta_t^g \sigma_{t-1}^g(s_t, \mathbf{x}_t), \beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t)\Big\} \ge \max\left\{\max_{(s,\mathbf{x}) \in G_t}\Big\{\beta_t^g \sigma_{t-1}^g(s, \mathbf{x})\Big\}, \max_{(s,\mathbf{x}) \in G_t \cup M_t}\Big\{\beta_t^f \sigma_{t-1}^f(s, \mathbf{x})\Big\}\right\}. \tag{47}$$

We bound the first term in (45) as follows:

$$\begin{aligned} f(s_*, \mathbf{x}_*) - f(s_t^{(\mathbf{x}_*)}, \mathbf{x}_*) &\le L_f|s_* - s_t^{(\mathbf{x}_*)}| \qquad \text{(by definition of } L_f) \\ &\le 2(L_f/L_g')\beta_t^g \sigma_{t-1}^g(s_t^{(\mathbf{x}_*)}, \mathbf{x}_*) \qquad \text{(by (44))} \\ &\le 2(L_f/L_g') \cdot \max\{\beta_t^g \sigma_{t-1}^g(s_t, \mathbf{x}_t), \beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t)\} \qquad \text{(by (47))}. \end{aligned} \tag{48}$$

We now consider the second term in (45). Since $\mathbf{x}_t$ was chosen instead of $\mathbf{x}_*$, we have that $\mathbf{x}_t$ was not eliminated. This implies that at least one of the two elimination criteria did not hold (note that both (20) and (21) must hold for $\mathbf{x}_t$ to be eliminated). If (20) did not hold, then we have

$$\begin{aligned} f(s_t^{(\mathbf{x}_*)}, \mathbf{x}_*) - f(\hat{s}_t^{(\mathbf{x}_t)}, \mathbf{x}_t) &\le \mathrm{UCB}_{t-1}^f(s_t^{(\mathbf{x}_*)}, \mathbf{x}_*) - \mathrm{LCB}_{t-1}^f(\hat{s}_t^{(\mathbf{x}_t)}, \mathbf{x}_t) \\ &= \mathrm{LCB}_{t-1}^f(s_t^{(\mathbf{x}_*)}, \mathbf{x}_*) + 2\beta_t^f \sigma_{t-1}^f(s_t^{(\mathbf{x}_*)}, \mathbf{x}_*) \\ &\qquad - \mathrm{UCB}_{t-1}^f(\hat{s}_t^{(\mathbf{x}_t)}, \mathbf{x}_t) + 2\beta_t^f \sigma_{t-1}^f(\hat{s}_t^{(\mathbf{x}_t)}, \mathbf{x}_t) \\ &\le 2\beta_t^f \sigma_{t-1}^f(s_t^{(\mathbf{x}_*)}, \mathbf{x}_*) + 2\beta_t^f \sigma_{t-1}^f(\hat{s}_t^{(\mathbf{x}_t)}, \mathbf{x}_t) \qquad \text{(by (20) being } \texttt{false}) \qquad (49) \\ &\le 4 \cdot \max\{\beta_t^g \sigma_{t-1}^g(s_t, \mathbf{x}_t), \beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t)\} \qquad \text{(by (47))} \qquad (50) \end{aligned}$$

Alternatively, if (21) did not hold, then we have

$$\begin{aligned}
f(s_t^{(\mathbf{x}_*)},\mathbf{x}_*) - f(s_t^{(\mathbf{x}_t)},\mathbf{x}_t) &\le \mathrm{UCB}^f_{t-1}(s_t^{(\mathbf{x}_*)},\mathbf{x}_*) - \mathrm{LCB}^f_{t-1}(s_t^{(\mathbf{x}_t)},\mathbf{x}_t) \\
&= \mathrm{LCB}^f_{t-1}(s_t^{(\mathbf{x}_*)},\mathbf{x}_*) + 2\beta_t^f\sigma^f_{t-1}(s_t^{(\mathbf{x}_*)},\mathbf{x}_*) \\
&\qquad - \mathrm{UCB}^f_{t-1}(s_t^{(\mathbf{x}_t)},\mathbf{x}_t) + 2\beta_t^f\sigma^f_{t-1}(s_t^{(\mathbf{x}_t)},\mathbf{x}_t) \\
&\le 2\beta_t^f\sigma^f_{t-1}(s_t^{(\mathbf{x}_*)},\mathbf{x}_*) + 2\beta_t^f\sigma^f_{t-1}(s_t^{(\mathbf{x}_t)},\mathbf{x}_t) + L_f|\underline{s}_t^{(\mathbf{x}_t)} - s_t^{(\mathbf{x}_t)}| \qquad \text{(by (21) being } \texttt{false}) \quad (51)\\
&\le 2\beta_t^f\sigma^f_{t-1}(s_t^{(\mathbf{x}_*)},\mathbf{x}_*) + 2\beta_t^f\sigma^f_{t-1}(s_t^{(\mathbf{x}_t)},\mathbf{x}_t) + 2(L_f/L'_g)\beta_t^g\sigma^g_t(s_t^{(\mathbf{x}_t)},\mathbf{x}_t) \qquad \text{(by (43))} \quad (52)\\
&\le 4\cdot\max\left\{\beta_t^g\sigma^g_{t-1}(s_t,\mathbf{x}_t),\beta_t^f\sigma^f_{t-1}(s_t,\mathbf{x}_t)\right\} \quad (53)\\
&\qquad + 2(L_f/L'_g)\max\left\{\beta_t^g\sigma^g_{t-1}(s_t,\mathbf{x}_t),\beta_t^f\sigma^f_{t-1}(s_t,\mathbf{x}_t)\right\} \qquad \text{(by (47))}. \quad (54)
\end{aligned}$$

Finally, we consider the third term in (45). Suppose again that $\mathbf{x}_t$ remained non-eliminated due to (20) being `false`. In this case, we have:

$$\begin{aligned}
f(s_t^{(\mathbf{x}_t)},\mathbf{x}_t) - f(s_t,\mathbf{x}_t) &\le \begin{cases} 0, & (\text{if } s_t = s_t^{(\mathbf{x}_t)}) \\ \mathrm{UCB}^f_{t-1}(s_t^{(\mathbf{x}_t)},\mathbf{x}_t) - \mathrm{LCB}^f_{t-1}(\hat{s}_t^{(\mathbf{x}_t)},\mathbf{x}_t), & (\text{if } s_t = \hat{s}_t^{(\mathbf{x}_t)}) \end{cases} \\
&\le \max\{0, \mathrm{UCB}^f_{t-1}(s_t^{(\mathbf{x}_t)},\mathbf{x}_t) - \mathrm{UCB}^f_{t-1}(\hat{s}_t^{(\mathbf{x}_t)},\mathbf{x}_t) + 2\beta_t^f\sigma^f_{t-1}(\hat{s}_t^{(\mathbf{x}_t)},\mathbf{x}_t)\} \\
&\le 2\beta_t^f\sigma^f_{t-1}(\hat{s}_t^{(\mathbf{x}_t)},\mathbf{x}_t) \quad (55)\\
&\le 2\cdot\max\{\beta_t^g\sigma^g_{t-1}(s_t,\mathbf{x}_t),\beta_t^f\sigma^f_{t-1}(s_t,\mathbf{x}_t)\} \qquad \text{(by (47))}, \quad (56)
\end{aligned}$$

where (55) holds because $\mathrm{UCB}^f_{t-1}(\hat{s}_t^{(\mathbf{x}_t)},\mathbf{x}_t) \ge \mathrm{UCB}^f_{t-1}(s_t^{(\mathbf{x}_t)},\mathbf{x}_t)$ by the definition of $\hat{s}_t^{(\mathbf{x}_t)}$ in (16). Regarding the first step above, we note that either $s_t = s_t^{(\mathbf{x}_t)}$ or $s_t = \hat{s}_t^{(\mathbf{x}_t)}$ must hold, because the actions chosen in any round must belong to either G_t or M_t.

Next, if $\mathbf{x}_t$ remained non-eliminated due to (21) being `false`, then we can bound the third term in (46) as follows:

$$\begin{aligned}
f(\hat{s}_t^{(\mathbf{x}_t)},\mathbf{x}_t) - f(s_t,\mathbf{x}_t) &\le \begin{cases} 0, & (\text{if } s_t = \hat{s}_t^{(\mathbf{x}_t)}) \\ \mathrm{UCB}^f_{t-1}(\hat{s}_t^{(\mathbf{x}_t)},\mathbf{x}_t) - \mathrm{LCB}^f_{t-1}(s_t^{(\mathbf{x}_t)},\mathbf{x}_t), & (\text{if } s_t = s_t^{(\mathbf{x}_t)}) \end{cases} \\
&\le \max\{0, \mathrm{LCB}^f_{t-1}(\hat{s}_t^{(\mathbf{x}_t)},\mathbf{x}_t) - \mathrm{UCB}^f_{t-1}(s_t^{(\mathbf{x}_t)},\mathbf{x}_t) \\
&\qquad + 2\beta_t^f\sigma^f_{t-1}(s_t^{(\mathbf{x}_t)},\mathbf{x}_t) + 2\beta_t^f\sigma^f_{t-1}(s_t^{(\mathbf{x}_t)},\mathbf{x}_t)\} \\
&\le \max\{0, L_f|\underline{s}_t^{(\mathbf{x}_t)} - s_t^{(\mathbf{x}_t)}| + 2\beta_t^f\sigma^f_{t-1}(s_t^{(\mathbf{x}_t)},\mathbf{x}_t) + 2\beta_t^f\sigma^f_{t-1}(s_t^{(\mathbf{x}_t)},\mathbf{x}_t)\} \qquad \text{(by (21) being } \texttt{false}) \\
&\le 2(L_f/L'_g)\beta_t^g\sigma^g_{t-1}(s_t^{(\mathbf{x}_t)},\mathbf{x}_t) + 2\beta_t^f\sigma^f_{t-1}(s_t^{(\mathbf{x}_t)},\mathbf{x}_t) + 2\beta_t^f\sigma^f_{t-1}(s_t^{(\mathbf{x}_t)},\mathbf{x}_t) \qquad \text{(by (43))} \quad (57)\\
&\le 2(L_f/L'_g)\max\{\beta_t^g\sigma^g_{t-1}(s_t,\mathbf{x}_t),\beta_t^f\sigma^f_{t-1}(s_t,\mathbf{x}_t)\} \quad (58)\\
&\qquad + 4\cdot\max\{\beta_t^g\sigma^g_{t-1}(s_t,\mathbf{x}_t),\beta_t^f\sigma^f_{t-1}(s_t,\mathbf{x}_t)\} \qquad \text{(by (47))}. \quad (59)
\end{aligned}$$

Combining the above three terms, we obtain the following bound on the *instantaneous regret* incurred by the algorithm (irrespective of whether $(s_t,\mathbf{x}_t) \in G_t$ or $(s_t,\mathbf{x}_t) \in M_t$, and irrespective of the condition that caused $\mathbf{x}_t$ to remain non-eliminated):

$$f(s_*,\mathbf{x}_*) - f(s_t,\mathbf{x}_t) \le \left(8 + \frac{6L_f}{L'_g}\right)\max\{\beta_t^g\sigma^g_{t-1}(s_t,\mathbf{x}_t),\beta_t^f\sigma^f_{t-1}(s_t,\mathbf{x}_t)\}. \quad (60)$$

A.1.2 Regret for $(s_*,\mathbf{x}_*) \in S_t$

Next, we consider $(s_*,\mathbf{x}_*) \in S_t$, i.e., $\mathrm{UCB}^g_{t-1}(s_*,\mathbf{x}_*) \le h$. In this case, it must hold that $s_* \le s_t^{(\mathbf{x}_*)}$ (since $s_t^{(\mathbf{x}_*)}$ is defined as the "highest" safe s for $\mathbf{x}_*$ in (14)). Thus, we have

$$\mathrm{UCB}^f_{t-1}(s_*,\mathbf{x}_*) \le \mathrm{UCB}^f_{t-1}(\hat{s}_t^{(\mathbf{x}_*)},\mathbf{x}_*), \quad (61)$$

since $\hat{s}_t^{(\mathbf{x}_*)}$ is the maximizer of $\mathrm{UCB}_{t-1}^f(\cdot, \mathbf{x}_*)$ over $s \le s_t^{(\mathbf{x}_*)}$. In this case, we can bound the instantaneous regret (45) directly as follows:

$$\begin{aligned}
f(s_*, \mathbf{x}_*) - f(s_t, \mathbf{x}_t) &\le \mathrm{UCB}_{t-1}^f(s_*, \mathbf{x}_*) - \mathrm{LCB}_{t-1}^f(s_t, \mathbf{x}_t) \\
&\le \mathrm{UCB}_{t-1}^f(\hat{s}_t^{(\mathbf{x}_*)}, \mathbf{x}_*) - \mathrm{UCB}_{t-1}^f(s_t, \mathbf{x}_t) + 2\beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t) \qquad \text{(by (61)} \\
&= \mathrm{LCB}_{t-1}^f(\hat{s}_t^{(\mathbf{x}_*)}, \mathbf{x}_*) - \mathrm{UCB}_{t-1}^f(s_t, \mathbf{x}_t) \\
&\qquad + 2\beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t) + 2\beta_t^f \sigma_{t-1}^f(\hat{s}_t^{(\mathbf{x}_*)}, \mathbf{x}_*) \qquad \text{(by (10))} \\
&\le L_f |\underline{s}_t^{(\mathbf{x}_t)} - s_t^{(\mathbf{x}_t)}| + 2\beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t) + 2\beta_t^f \sigma_{t-1}^f(\hat{s}_t^{(\mathbf{x}_*)}, \mathbf{x}_*) \qquad \text{(see (64) below)} \qquad (62) \\
&\le 2(L_f/L_g')\beta_t^g \sigma_{t-1}^g(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t) + 2\beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t) + 2\beta_t^f \sigma_{t-1}^f(\hat{s}_t^{(\mathbf{x}_*)}, \mathbf{x}_*) \qquad \text{(by (43))} \\
&\le 2(L_f/L_g') \max\left\{\beta_t^g \sigma_{t-1}^g(s_t, \mathbf{x}_t), \beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t)\right\} \\
&\qquad + 4 \cdot \max\left\{\beta_t^g \sigma_{t-1}^g(s_t, \mathbf{x}_t), \beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t)\right\} \qquad \text{(by (47))}, \qquad (63)
\end{aligned}$$

where (62) holds because of the following:

$$\begin{aligned}
&\mathrm{LCB}_{t-1}^f(\hat{s}_t^{(\mathbf{x}_*)}, \mathbf{x}_*) - \mathrm{UCB}_{t-1}^f(s_t, \mathbf{x}_t) \\
&\qquad \le \begin{cases} \max\left\{0, L_f |\underline{s}_t^{(\mathbf{x}_t)} - s_t^{(\mathbf{x}_t)}|\right\}, \text{ if } s_t = \hat{s}_t^{(\mathbf{x}_t)} \quad (\text{since } \mathtt{elim}_t(\mathbf{x}_t) = \mathtt{false})) \\ L_f |\underline{s}_t^{(\mathbf{x}_t)} - s_t^{(\mathbf{x}_t)}|, \text{ if } s_t = s_t^{(\mathbf{x}_t)} \quad (\text{since } \mathtt{expd}_t(\mathbf{x}_t) = \mathtt{true}) \ . \end{cases} \qquad (64)
\end{aligned}$$

Here, we again use the fact that either $s_t = \hat{s}_t^{(\mathbf{x}_t)}$ or $s_t = s_t^{(\mathbf{x}_t)}$ (as the action $(s_t, \mathbf{x}_t)$ chosen at round t must belong to either M_t or G_t). Therefore, either (20) or (21) must be `false` when $s_t = \hat{s}_t^{(\mathbf{x}_t)}$ (since $\mathbf{x}_t$ was not eliminated), and (22) must be `true` when $s_t = s_t^{(\mathbf{x}_t)}$ (since an action on the safe boundary is only chosen if it considered "potentially beneficial" to expand, as decided by $\mathtt{expd}_t$).

A.1.3 Bounding the cumulative regret

Summing the instantaneous regret terms in (60) and (63) from $t = 1, \ldots, T$, we get the following:

$$\begin{aligned}
R_T = \sum_{t=1}^T (f(s_*, \mathbf{x}_*) - f(s_t, \mathbf{x}_t)) &\le \left(8 + \frac{6L_f}{L_g'}\right) \sum_{t=1}^T \max\left\{\beta_t^g \sigma_{t-1}^g(s_t, \mathbf{x}_t), \beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t)\right\} \qquad (65) \\
&\le \left(8 + \frac{6L_f}{L_g'}\right) \sum_{t=1}^T \left(\beta_t^g \sigma_{t-1}^g(s_t, \mathbf{x}_t) + \beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t)\right) \qquad (66) \\
&\le \left(8 + \frac{6L_f}{L_g'}\right) \left(\beta_T^g \sum_{t=1}^T \sigma_{t-1}^g(s_t, \mathbf{x}_t) + \beta_T^f \sum_{t=1}^T \sigma_{t-1}^f(s_t, \mathbf{x}_t)\right), \qquad (67)
\end{aligned}$$

where we used the assumption that β_t^f, β_t^g are non-decreasing with respect to t. Then, from Lemma 4 of (Chowdhury and Gopalan, 2017), we have the standard bounds:

$$\sum_{t=1}^T \sigma_{t-1}^g(s_t, \mathbf{x}_t) = O\big(\sqrt{T\gamma_T^g}\big), \qquad (68)$$

$$\sum_{t=1}^T \sigma_{t-1}^f(s_t, \mathbf{x}_t) = O\big(\sqrt{T\gamma_T^f}\big). \qquad (69)$$

Hence, with probability at least $1 - \delta$,

$$R_T = O\left(\left(1 + \frac{L_f}{L_g'}\right)\left(\beta_T^g \sqrt{T\gamma_T^g} + \beta_T^f \sqrt{T\gamma_T^f}\right)\right). \qquad (70)$$

Specifically, for the choice of β_f and β_g from Lemma 1, we have the following:

$$R_T = O\Bigg(B_g\left(1+\frac{L_f}{L'_g}\right)\sqrt{T\gamma_T^g} + R_g\left(1+\frac{L_f}{L'_g}\right)\sqrt{T\gamma_T^g\left(\gamma_T^g+\ln(1/\delta)\right)} \\ + B_f\left(1+\frac{L_f}{L'_g}\right)\sqrt{T\gamma_T^f} + R_f\left(1+\frac{L_f}{L'_g}\right)\sqrt{T\gamma_T^f\left(\gamma_T^f+\ln(1/\delta)\right)}\Bigg), \tag{71}$$

where $\beta_T^g \leq B_g + R_g\sqrt{2\left(\gamma_T^g+1+\ln(2/\delta)\right)}$ since γ_t^g is monotonically increasing (and similarly for β_T^f).

A.2 Proof of Theorem 1 (Case 3)

This case turns out to be a minor variation of the above, so we omit the full details and only describe the differences. First, we note that the initial results derived with respect to the safety function g hold in this case as well. Specifically, (39) to (44) are valid because of the same arguments as presented in the previous proof.

Next, (47) also holds given the modified acquisition function for case 3, with the only change being that the set M_t is not considered here. Also, note that the optimal safe action is guaranteed to lie on the "safe boundary" in this case. We first consider the scenario where $(s_*, \mathbf{x}_*) \notin S_t$, which implies that (48) holds without modification.

Since the elimination criteria is simplified, following (50), we now have

$$f(s_t^{(\mathbf{x}_*)}, \mathbf{x}_*) - f(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t) \leq \left(4+\frac{2L_f}{L'_g}\right)\max\left\{\beta_t^g\sigma_{t-1}^g(s_t,\mathbf{x}_t), \beta_t^f\sigma_{t-1}^f(s_t,\mathbf{x}_t)\right\}. \tag{72}$$

Finally, since $s_t = s_t^{(\mathbf{x}_t)}$ for all $t \geq 1$ in this case, we can combine (48) and (72) to obtain a bound on the instantaneous regret as follows:

$$f(s_*, \mathbf{x}_*) - f(s_t, \mathbf{x}_t) \leq \left(4+\frac{4L_f}{L'_g}\right)\max\left\{\beta_t^g\sigma_{t-1}^g(s_t,\mathbf{x}_t), \beta_t^f\sigma_{t-1}^f(s_t,\mathbf{x}_t)\right\}. \tag{73}$$

It may also hold that $(s_*, \mathbf{x}_*) \in S_t$ when $s_* = 1$. In this case, we bound the instantaneous regret directly as follows:

$$\begin{aligned} f(s_*,\mathbf{x}_*) - f(s_t,\mathbf{x}_t) &\leq \mathrm{UCB}_{t-1}^f(s_*,\mathbf{x}_*) - \mathrm{LCB}_{t-1}^f(s_t,\mathbf{x}_t) \\ &= \mathrm{UCB}_{t-1}^f(s_*,\mathbf{x}_*) - \mathrm{UCB}_{t-1}^f(s_t,\mathbf{x}_t) + 2\beta_t^f\sigma_{t-1}^f(s_t,\mathbf{x}_t) \\ &= \mathrm{LCB}_{t-1}^f(s_*,\mathbf{x}_*) - \mathrm{UCB}_{t-1}^f(s_t,\mathbf{x}_t) + 2\beta_t^f\sigma_{t-1}^f(s_t,\mathbf{x}_t) + 2\beta_t^f\sigma_{t-1}^f(s_*,\mathbf{x}_*) \\ &\leq 2\beta_t^f\sigma_{t-1}^f(s_t,\mathbf{x}_t) + 2\beta_t^f\sigma_{t-1}^f(s_*,\mathbf{x}_*) \qquad \text{(by (21) being false, and since } s_t^{(\mathbf{x}_*)} = \underline{s}_t^{(\mathbf{x}_*)}) \\ &\leq 4\cdot\max\left\{\beta_t^g\sigma_{t-1}^g(s_t,\mathbf{x}_t), \beta_t^f\sigma_{t-1}^f(s_t,\mathbf{x}_t)\right\} \qquad \text{(by (47))}. \end{aligned} \tag{74}$$

The cumulative regret bound follows similarly to the previous proof by summing over the instantaneous regret terms in (73) and (74).

A.3 Proof of Theorem 2

Similar to the proof of Theorem 1, we split the proof into two possible scenarios based on the chosen $\mathbf{x}_t$ in each round: (i) $(s_*^{(\mathbf{x}_t)}, \mathbf{x}_t) \notin S_t$, i.e., $\mathrm{UCB}_{t-1}^g(s_*^{(\mathbf{x}_t)}, \mathbf{x}_t) > h$ and (ii) $(s_*^{(\mathbf{x}_t)}, \mathbf{x}_t) \in S_t$, i.e., $\mathrm{UCB}_{t-1}^g(s_*^{(\mathbf{x}_t)}, \mathbf{x}_t) \leq h$.

A.3.1 Regret for $(s_*^{(\mathbf{x}_t)}, \mathbf{x}_t) \notin S_t$

Following the proof of Theorem 1, several results concerning the safety function g hold. Specifically, the inequalities not concerning $\mathbf{x}_*$ from (40) to (43) continue to hold for the modified algorithm, due to the same arguments as presented earlier (with (41) skipped). However, for finding the regret r'_t here, we now consider the optimal s corresponding to $\mathbf{x}_t$ (i.e., $s_*^{(\mathbf{x}_t)}$) as defined in (2), instead of the global safe optimum action. The inequalities derived earlier are modified accordingly as follows.

Corresponding to (39), we now have

$$g(s_*^{(\mathbf{x}_t)}, \mathbf{x}_t) \leq h. \tag{75}$$

Moreover, from (75) and (40), we have:

$$\begin{aligned}
L'_g|s_*^{(\mathbf{x}_t)} - s_t^{(\mathbf{x}_t)}| &\leq g(s_*^{(\mathbf{x}_t)}, \mathbf{x}_t) - g(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t) \qquad \text{(by the definition of } L'_g) \\
&\leq h - \mathrm{LCB}^g_{t-1}(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t) \qquad \text{(by (75))} \\
&\leq \mathrm{UCB}^g_{t-1}(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t) - \mathrm{LCB}^g_{t-1}(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t) \qquad \text{(by (40)} \\
&= 2\beta^g_t \sigma^g_{t-1}(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t) \\
\implies |s_*^{(\mathbf{x}_t)} - s_t^{(\mathbf{x}_t)}| &\leq 2\beta^g_t \sigma^g_{t-1}(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t)/L'_g.
\end{aligned} \tag{76}$$

Next, with respect to the objective function f, we have

$$\begin{aligned}
f(s_*^{(\mathbf{x}_t)}, \mathbf{x}_t) - f(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t) &\leq L_f|s_*^{(\mathbf{x}_t)} - s_t^{(\mathbf{x}_t)}| \qquad \text{(by definition of } L_f) \\
&\leq 2(L_f/L'_g)\beta^g_t \sigma^g_{t-1}(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t) \qquad \text{(by (76))}.
\end{aligned} \tag{77}$$

In addition, we have the following:

$$f(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t) - f(s_t, \mathbf{x}_t) \leq \begin{cases} 0, & (\text{if } s_t = s_t^{(\mathbf{x}_t)}) \\ \mathrm{UCB}^f_{t-1}(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t) - \mathrm{LCB}^f_{t-1}(\hat{s}_t^{(\mathbf{x}_t)}, \mathbf{x}_t), & (\text{if } s_t = \hat{s}_t^{(\mathbf{x}_t)}, \text{ by (28)}) \end{cases} \tag{78}$$

$$\leq \mathrm{UCB}^f_{t-1}(\hat{s}_t^{(\mathbf{x}_t)}, \mathbf{x}_t) - \mathrm{LCB}^f_{t-1}(\hat{s}_t^{(\mathbf{x}_t)}, \mathbf{x}_t) \tag{79}$$

$$\leq 2\beta^f_t \sigma^f_{t-1}(\hat{s}_t^{(\mathbf{x}_t)}, \mathbf{x}_t), \tag{80}$$

where we again use the fact that the chosen action must belong to either G_t (i.e, $s_t = s_t^{(\mathbf{x}_t)}$) or M_t (i.e., $s_t = \hat{s}_t^{(\mathbf{x}_t)}$). Adding (77) and (80), we obtain

$$\begin{aligned}
f(s_*^{(\mathbf{x}_t)}, \mathbf{x}_t) - f(s_t, \mathbf{x}_t) &\leq 2(L_f/L'_g)\beta^g_t \sigma^g_{t-1}(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t) + 2\beta^f_t \sigma^f_{t-1}(\hat{s}_t^{(\mathbf{x}_t)}, \mathbf{x}_t) \\
&\leq \left(2 + \frac{2L_f}{L'_g}\right) \max\{\beta^g_t \sigma^g_{t-1}(s_t, \mathbf{x}_t), \beta^f_t \sigma^f_{t-1}(s_t, \mathbf{x}_t)\}.
\end{aligned} \tag{81}$$

A.3.2 Regret for $(s_*^{(\mathbf{x}_t)}, \mathbf{x}_t) \in S_t$

When $(s_*^{(\mathbf{x}_t)}, \mathbf{x}_t) \in S_t$, similarly to earlier as in (61), we have:

$$\mathrm{UCB}^f_{t-1}(s_*^{(\mathbf{x}_t)}, \mathbf{x}_t) \leq \mathrm{UCB}^f_{t-1}(\hat{s}^{(\mathbf{x}_t)}, \mathbf{x}_t). \tag{82}$$

In this case, we can bound the instantaneous regret $r'_t = f(s_*^{(\mathbf{x}_t)}, \mathbf{x}_t) - f(s_t, \mathbf{x}_t)$ directly as follows:

$$\begin{aligned}
f(s_*^{(\mathbf{x}_t)}, \mathbf{x}_t) - f(s_t, \mathbf{x}_t) &\leq \mathrm{UCB}^f_{t-1}(s_*^{(\mathbf{x}_t)}, \mathbf{x}_t) - \mathrm{LCB}^f_{t-1}(s_t, \mathbf{x}_t) \qquad \text{(by (28))} \\
&\leq \begin{cases} 2\beta^f_t \sigma^f_{t-1}(\hat{s}^{(\mathbf{x}_t)}, \mathbf{x}_t), \text{ if } s_t = \hat{s}_t^{(\mathbf{x}_t)}, & \text{(by (82))} \\ \mathrm{UCB}^f_{t-1}(s_*^{(\mathbf{x}_t)}, \mathbf{x}_t) - \mathrm{LCB}^f_{t-1}(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t), \text{ if } s_t = s_t^{(\mathbf{x}_t)}. \end{cases}
\end{aligned} \tag{83}$$

Furthermore, we have the following when $s_t = s_t^{(\mathbf{x}_t)}$:

$$\mathrm{UCB}^f_{t-1}(s_*^{(\mathbf{x}_t)}, \mathbf{x}_t) - \mathrm{LCB}^f_{t-1}(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t) \leq \mathrm{LCB}^f_{t-1}(s_*^{(\mathbf{x}_t)}, \mathbf{x}_t) - \mathrm{UCB}^f_{t-1}(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t) \tag{84}$$

$$2\beta^f_t \sigma^f_{t-1}(s_*^{(\mathbf{x}_t)}, \mathbf{x}_t) + 2\beta^f_t \sigma^f_{t-1}(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t) \tag{85}$$

$$\leq L_f|\underline{s}_t^{(\mathbf{x}_t)} - s_t^{(\mathbf{x}_t)}| + 2\beta^f_t \sigma^f_{t-1}(s_*^{(\mathbf{x}_t)}, \mathbf{x}_t) + 2\beta^f_t \sigma^f_{t-1}(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t) \qquad \text{(by (24))} \tag{86}$$

$$\leq 2(L_f/L'_g)\beta^g_t \sigma^g_{t-1}(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t) + 2\beta^f_t \sigma^f_{t-1}(s_*^{(\mathbf{x}_t)}, \mathbf{x}_t) + 2\beta^f_t \sigma^f_{t-1}(s_t^{(\mathbf{x}_t)}, \mathbf{x}_t) \qquad \text{(by (43))} \tag{87}$$

$$\leq 2\left((L_f/L'_g) + 2\right) \max\{\beta^g_t \sigma^g_{t-1}(s_t, \mathbf{x}_t), \beta^f_t \sigma^f_{t-1}(s_t, \mathbf{x}_t)\} \qquad \text{(by (47))}. \tag{88}$$

Combining this with (83), we obtain

$$f(s_*^{(\mathbf{x}_t)}, \mathbf{x}_t) - f(s_t, \mathbf{x}_t) \leq 2\left((L_f/L'_g) + 2\right) \max\{\beta_t^g \sigma_{t-1}^g(s_t, \mathbf{x}_t), \beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t)\}. \tag{89}$$

Finally, combining the results from $t = 1, \ldots, T$, we can conclude that

$$R'_T = \sum_{t=1}^{T} r'_t = \sum_{t=1}^{T} \left(f(s_*^{(\mathbf{x}_t)}, \mathbf{x}_t) - f(s_t, \mathbf{x}_t)\right) \tag{90}$$

$$\leq 2\left((L_f/L'_g) + 2\right) \max\left\{\beta_t^g \sigma_{t-1}^g(s_t, \mathbf{x}_t), \beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t)\right\}. \tag{91}$$

As before, we can use Lemma 4 from (Chowdhury and Gopalan, 2017) to conclude that with probability at least $1 - \delta$,

$$R'_T = O\left(\left(1 + \frac{L_f}{L'_g}\right)\left(\beta_T^g \sqrt{T\gamma_T^g} + \beta_T^f \sqrt{T\gamma_T^f}\right)\right). \qquad \square$$

A.4 Proof of Theorem 3

All results concerning the safety function g from the previous theorem's proof hold here. However, when considering the objective function f, note that we are no longer concerned with the instantaneous regret incurred with respect to the action chosen by the algorithm. Instead, we now consider the best estimate $\hat{s}_t^{(\mathbf{x})}$ for any given $\mathbf{x}$ at round t, and try to bound the worst-case (over $\mathbf{x} \in \mathcal{D}_{\mathcal{X}}$) simple regret incurred by the algorithm.

Suppose that the worst-case simple regret is incurred by $\mathbf{x} = \underline{\mathbf{x}}_t$ at round t, i.e.,

$$\underline{\mathbf{x}}_t = \operatorname*{arg\,max}_{\mathbf{x} \in \mathcal{D}_{\mathcal{X}}^t} \left\{f(s_*^{(\mathbf{x})}, \mathbf{x}) - f(\hat{s}_t^{(\mathbf{x})}, \mathbf{x})\right\}. \tag{92}$$

We again proceed to split the analysis into two cases.

A.4.1 Regret for $(s_*^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) \notin S_t$

First, we consider $(s_*^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) \notin S_t$. In this case,

$$f(s_*^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) - f(s_t^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) \leq L_f |s_*^{(\underline{\mathbf{x}}_t)} - s_t^{(\underline{\mathbf{x}}_t)}| \qquad \text{(by definition of } L_f) \tag{93}$$

$$\leq 2(L_f/L'_g)\beta_t^g \sigma_{t-1}^g(s_t^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) \qquad \text{(by (76))}. \tag{94}$$

Next, we derive the following:

$$f(s_t^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) - f(\hat{s}_t^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) \leq \mathrm{UCB}_{t-1}^f(s_t^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) - \mathrm{LCB}_{t-1}^f(\hat{s}_t^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) \tag{95}$$

$$\leq \mathrm{UCB}_{t-1}^f(s_t^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) - \mathrm{UCB}_{t-1}^f(\hat{s}_t^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) + 2\beta_t^f \sigma_{t-1}^f(\hat{s}_t^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) \tag{96}$$

$$\leq 2\beta_t^f \sigma_{t-1}^f(\hat{s}_t^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) \qquad \text{(by (16))}. \tag{97}$$

Adding (94) and (97), we have a bound on the worst-case simple regret incurred at round t as follows:

$$f(s_*^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) - f(\hat{s}_t^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) \leq 2(L_f/L'_g)\beta_t^g \sigma_{t-1}^g(s_t^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) + 2\beta_t^f \sigma_{t-1}^f(\hat{s}_t^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t). \tag{98}$$

A.4.2 Regret for $(s_*^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) \in S_t$

When $(s_*^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) \in S_t$, we can bound the instantaneous regret $r_t^{\mathcal{X}}$ directly, as follows:

$$r_t^{\mathcal{X}} = f(s_*^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) - f(\hat{s}_t^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) \leq \mathrm{UCB}_{t-1}^f(s_*^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) - \mathrm{LCB}_{t-1}^f(\hat{s}_t^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) \tag{99}$$

$$\leq \mathrm{UCB}_{t-1}^f(s_*^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) - \mathrm{UCB}_{t-1}^f(\hat{s}_t^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) + 2\beta_t^f \sigma_{t-1}^f(\hat{s}_t^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) \tag{100}$$

$$\leq 2\beta_t^f \sigma_{t-1}^f(\hat{s}_t^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) \qquad \text{(by (16))}. \tag{101}$$

Therefore, considering the cumulative worst-case simple regret for $t = 1, \dots, T$, we have:

$$R_T^{\mathcal{X}} = \sum_{t=1}^{T} r_t^{\mathcal{X}} = \sum_{t=1}^{T} \Big(f(s_*^{(\mathbf{x}_t)}, \mathbf{x}_t) - f(\hat{s}_t, \mathbf{x}_t) \Big) \tag{102}$$

$$\leq 2(L_f/L_g')\beta_t^g \sigma_{t-1}^g(s_t^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) + 2\beta_t^f \sigma_{t-1}^f(\hat{s}_t^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) \tag{103}$$

$$\leq 2\left((L_f/L_g') + 1\right) \max\{\beta_t^g \sigma_{t-1}^g(s_t, \mathbf{x}_t), \beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t)\}. \tag{104}$$

Once again, we can use Lemma 4 from (Chowdhury and Gopalan, 2017) to conclude that with probability at least $1 - \delta$,

$$R_T^{\mathcal{X}} = O\left(\left(1 + \frac{L_f}{L_g'} \right) \left(\beta_T^g \sqrt{T\gamma_T^g} + \beta_T^f \sqrt{T\gamma_T^f} \right) \right). \tag{105}$$

A.5 Refining the Regret Bounds

In this section, we discuss an alternative acquisition function, defined as:

$$\mathtt{acq}_t(s, \mathbf{x}) = \max\{\beta_t^f \sigma_{t-1}^f(s, \mathbf{x}), (L_f/L_g')\beta_t^g \sigma_{t-1}^g(s, \mathbf{x})\}, \text{, if } (s, \mathbf{x}) \in G_t. \tag{106}$$

Note that we only redefine the function for $(s, \mathbf{x}) \in G_t$, while the function remains unchanged for the other cases (as stated in (23) and (25)). This acquisition function provides certain desirable scaling properties of our theoretical guarantees, as discussed below.

We note that the proofs of all theorems eventually used the acquisition function for establishing bounds of the following form (such as in (50), (63) and (74)):

$$\beta_t^f \sigma_{t-1}^f(\mathbf{z}) \leq \max\{\beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t), \beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t)\}, \tag{107}$$

$$(L_f/L_g')\beta_t^g \sigma_{t-1}^g(\mathbf{z}) \leq (L_f/L_g') \max\{\beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t), \beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t)\}, \tag{108}$$

where $\mathbf{z}$ denotes some safe action (depending on the corresponding equation in our earlier proofs), while the other steps of the proofs did not depend on the acquisition function's definition. The $\max\{\beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t), \beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t)\}$ terms were then upper bounded by their sum (such as in (66)) as follows:

$$\max\{\beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t), \beta_t^g \sigma_{t-1}^g(s_t, \mathbf{x}_t)\} \leq \beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t) + \beta_t^g \sigma_{t-1}^g(s_t, \mathbf{x}_t), \tag{109}$$

$$(L_f/L_g') \max\{\beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t), \beta_t^g \sigma_{t-1}^g(s_t, \mathbf{x}_t)\} \leq (L_f/L_g')\beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t) + (L_f/L_g')\beta_t^g \sigma_{t-1}^g(s_t, \mathbf{x}_t). \tag{110}$$

Note that these steps introduced the factor (L_f/L_g') into the f-terms as well, resulting in the regret bounds stated in our theorems.

With the alternative acquisition function given in (36), our proofs can instead use the following refined steps:

$$\beta_t^f \sigma_{t-1}^f(\mathbf{z}) \leq \max\{\beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t), (L_f/L_g')\beta_t^g \sigma_{t-1}^g(s_t, \mathbf{x}_t)\}, \tag{111}$$

$$(L_f/L_g')\beta_t^g \sigma_{t-1}^g(\mathbf{z}) \leq \max\{\beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t), (L_f/L_g')\beta_t^g \sigma_{t-1}^g(s_t, \mathbf{x}_t)\}. \tag{112}$$

Again using the fact that the maximum of the two terms is upper bounded by their sum, we can derive the following:

$$\beta_t^f \sigma_{t-1}^f(\mathbf{z}) \leq \beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t) + (L_f/L_g')\beta_t^g \sigma_{t-1}^g(s_t, \mathbf{x}_t), \tag{113}$$

$$(L_f/L_g')\beta_t^g \sigma_{t-1}^g(\mathbf{z}) \leq \beta_t^f \sigma_{t-1}^f(s_t, \mathbf{x}_t) + (L_f/L_g')\beta_t^g \sigma_{t-1}^g(s_t, \mathbf{x}_t). \tag{114}$$

Therefore, when applying Lemma 4 from (Chowdhury and Gopalan, 2017), the (L_f/L_g') factor only gets multiplied with the $\beta_T^g \sqrt{T\gamma_T^g}$ term, and not the $\beta_T^f \sqrt{T\gamma_T^f}$ term. Hence, we get the following regret bound for R_T in case 1 and case 3 (similarly for R_T' and $R_T^{\mathcal{X}}$ in case 2):

$$R_T = O\left(\beta_T^f \sqrt{T\gamma_T^f} + \frac{L_f}{L_g'} \beta_T^g \sqrt{T\gamma_T^g} \right). \tag{115}$$

Behavior with respect to rescaling: To motivate the notion of rescaling f and/or g, we argue a certain equivalence between the following two problems for arbitrary $c > 0$:

(i) function f with RKHS norm B_f and noise level R_f;

(ii) function cf with RKHS norm cB_f and noise level cR_f.

The equivalence comes from the fact that observations y_t^f for problem (ii) can be obtained from those for problem (i) by simply multiplying by c (or vice versa by dividing). Hence, any algorithm for problem (ii) can be applied to problem (i), and vice versa. The only difference is that in problem (ii) the regret is scaled by c compared to problem (i). In contrast, if g, B_g, and R_g are similarly scaled (as well as h) then a similar equivalence holds without any change in the regret (which is measured only with respect to f).

In accordance with this discussion, a regret bound should ideally scale linearly when f (and B_f, R_f) is scaled, and should stay unchanged when g is scaled. This behavior is indeed observed in the refined regret bounds that we derived above: Scaling f by c results in scaling β_T^f and L_f by c, thereby scaling the regret bound by c as well. However, scaling g scales β_T^g and L'_g by c, and thus, the regret bound remains unchanged due to cancellation of the factor c. Although this scaling property of the regret bounds holds with the modified acquisition function, it comes at the cost of an additional dependence of the algorithm on L_f and L'_g. Since this may not be desirable in practice due to unavailability of good estimates of these constants, we use the original forms of the acquisition in our theorems and experiments.

A.6 Recovering the Guarantees of M-SAFEUCB

As noted in Section 3 under Case 3, modifying the functions $\texttt{elim}_t$, $\texttt{expd}_t$ and $\texttt{acq}_t$ suitably gives us the M-SafeUCB algorithm of Losalka and Scarlett (2023) for the goal of finding the optimal s for every $\mathbf{x} \in \mathcal{D}_{\mathcal{X}}$, when both f and g are known to be monotone with respect to s. Specifically, Theorem 1 in (Losalka and Scarlett, 2023) gives a bound on a suitably-defined notion of cumulative regret, and Theorem 2 in (Losalka and Scarlett, 2023) states that the entire safe boundary will be accurately identified after a certain number of rounds.

To achieve this goal, we set $\texttt{elim}_t(\mathbf{x}) = \texttt{false}$ for every $\mathbf{x}$, set $\texttt{expd}_t(\mathbf{x}) = \texttt{true}$ for every $\mathbf{x}$ (except when $s_t^{(\mathbf{x})} = 1$), and set the acquisition function to be the same as that in (25), except that here we only use σ_{t-1}^g here and omit σ_{t-1}^f. Note that these choices imply that the algorithm does not need to use the constants L_f and L'_g.

We suppose that the algorithm is run using only the safety function g, which is reasonable since for each $\mathbf{x}$, the best s for f is the same as the best s for g (since both are monotone), namely, $s_*^{(\mathbf{x})} = \max\{s \in \mathcal{D}_{\mathcal{S}} : g(s, \mathbf{x}) \le h\}$. Furthermore, since we do not consider f in the algorithm in this case, we define $\hat{s}_t^{(\mathbf{x})}$ as the safe maximizer of $\mathrm{UCB}_{t-1}^g(\cdot, \mathbf{x})$ instead of $\mathrm{UCB}_{t-1}^f(\cdot, \mathbf{x})$.

Attaining Theorem 1 of (Losalka and Scarlett, 2023). Referring back to the proof for Theorem 2, (75) holds for all $\mathbf{x} \in \mathcal{D}_{\mathcal{X}}$ by the definition of the optimal safe s, i.e., $s_*^{(\mathbf{x})}$. Also, note that $(s_*^{(\mathbf{x})}, \mathbf{x}) \notin S_t$ in this case for any $\mathbf{x} \in \mathcal{D}_{\mathcal{X}}$ (when the confidence bounds are valid), since the optimal actions lie on the "safe boundary" (due to the monotonicity of both f and g). Therefore, the following can be derived by similar reasoning as that used in (76):

$$\begin{aligned} r_t^g &:= g(s_*^{(\mathbf{x}_t)}, \mathbf{x}_t) - g(s_t, \mathbf{x}_t) \\ &\le h - \mathrm{LCB}_{t-1}^g(s_t, \mathbf{x}_t) \qquad \text{(by (75))} \\ &\le \mathrm{UCB}_{t-1}^g(s_t, \mathbf{x}_t) - \mathrm{LCB}_{t-1}^g(s_t, \mathbf{x}_t) \qquad \text{(by (40)} \\ &= 2\beta_t^g \sigma_{t-1}^g(s_t, \mathbf{x}_t), \end{aligned} \tag{116}$$

where r_t^g is the instantaneous regret incurred by the algorithm by choosing $(s_t, \mathbf{x}_t)$ at round t. Note that $s_t = s_t^{(\mathbf{x}_t)}$ for every t in this case (since all actions must be chosen from the set G_t), and $\mathrm{UCB}_{t-1}^g(s_t, \mathbf{x}_t) < h$ does not hold (since any $(s_t^{(\mathbf{x})}, \mathbf{x})$ such that $s_t^{(\mathbf{x})} = 1$ is never selected with the redefined $\texttt{expd}_t$). Also note that the problem setting of (Losalka and Scarlett, 2023) precludes the scenario where g is safe over the entire input domain $\mathcal{D}$; thus, excluding actions with $s_t^{(\mathbf{x})} = 1$ does not lead to the situation that the algorithm is unable to chose any action in some round t.

In the case that $f = g$, we readily obtain Theorem 1 from (Losalka and Scarlett, 2023) by summing r_t^g over $t = 1, \dots, T$, and using Lemma 4 from (Chowdhury and Gopalan, 2017). Similar to the setup in (Losalka and Scarlett, 2023), this derivation does not require $L'_g > 0$ (i.e., g only needs to be non-decreasing rather than strictly increasing in this case). A similar argument also applies in the case that $f \neq g$ and both f and g are monotone with respect to s, but in this more general case, we need to that assume $L'_g > 0$ (and $L_f < \infty$) for the reasons discussed in Section C.1, and the final regret bound incurs a $1 + \frac{L_f}{L'_g}$ factor in the same way as Theorems 1–3. The algorithm itself does not need to know L_f and L'_g.

Attaining Theorem 2 of (Losalka and Scarlett, 2023). We refer to the proof of our Theorem 3 and consider the regret analysis for $(s_*^{(\underline{\mathbf{x}}_t)}, \underline{\mathbf{x}}_t) \notin S_t$ only (for the same reason as above). Note that given the design of the algorithm, we have $s_t = s_t^{(\mathbf{x})} = \hat{s}_t^{(\mathbf{x})}$, i.e., for any $\mathbf{x}$, the s on the current "safe boundary" is also the one that maximizes UCB_{t-1}^g.

Furthermore, we note that if $s_t^{(\mathbf{x})} = 1$ for any $\mathbf{x} \in \mathcal{D}_{\mathcal{X}}$, that would imply that for this specific value of $\mathbf{x}$, $(s, \mathbf{x})$ is safe for every $s \in \mathcal{D}_{\mathcal{S}}$ (based on validity of our confidence bounds). Accordingly, in the following, we only consider $\mathbf{x} \in \mathcal{D}_{\mathcal{X}}$ for which $s_t^{(\mathbf{x})} < 1$.

Based on the discussion above, the worst-case (with respect to $\mathbf{x}$) suboptimality of the algorithm's best guess (of the optimal s) can be bounded as follows:

$$\begin{aligned}
&\max_{\mathbf{x}: s_t^{(\mathbf{x})} < 1} \Big\{ g(s_*^{(\mathbf{x})}, \mathbf{x}) - g(\hat{s}_t^{(\mathbf{x})}, \mathbf{x}) \Big\} \\
&\quad \le \max_{\mathbf{x}: s_t^{(\mathbf{x})} < 1} \Big\{ h - \mathrm{LCB}(\hat{s}_t^{(\mathbf{x})}, \mathbf{x}) \Big\} \\
&\quad \le \max_{\mathbf{x}: s_t^{(\mathbf{x})} < 1} \Big\{ \underline{\mathrm{UCB}}(s_t^{(\mathbf{x})}, \mathbf{x}) - \mathrm{LCB}(\hat{s}_t^{(\mathbf{x})}, \mathbf{x}) \Big\} \quad (\text{since } \underline{\mathrm{UCB}}(s_t^{(\mathbf{x})}, \mathbf{x}) \ge h \text{ when } s_t^{(\mathbf{x})} < 1) \\
&\quad \le \max_{\mathbf{x}: s_t^{(\mathbf{x})} < 1} \Big\{ 2\beta_t^g \sigma_{t-1}^g(s_t^{(\mathbf{x})}, \mathbf{x}) \Big\} \quad (\text{since } \hat{s}_t^{(\mathbf{x})} = s_t^{(\mathbf{x})}, \text{ and } \underline{\mathrm{UCB}}(s_t^{(\mathbf{x})}, \mathbf{x}) \le \mathrm{UCB}(s_t^{(\mathbf{x})}, \mathbf{x})) \\
&\quad = 2\beta_t^g \sigma_{t-1}^g(s_t, \mathbf{x}_t) \quad (\text{since } \mathtt{acq}_t \text{ maximizes } \sigma_{t-1}^g(s, \mathbf{x}) \text{ over } (s, \mathbf{x}) \in G_t). \qquad (117)
\end{aligned}$$

As earlier, applying Lemma 4 from Chowdhury and Gopalan (2017) upper bounds the cumulative value of the worst-case regret derived in (117). Doing so essentially recovers Theorem 2 of Losalka and Scarlett (2023) (with $f = g$), with the difference that their result is not based on a cumulative measure, but rather based on returning the best estimate of s for each $\mathbf{x}$ *after* all queries have been taken. However, the latter follows as a simple consequence of the former by letting the final $\hat{s}_t^{(\mathbf{x})}$ be the maximum among $\{\hat{s}_t^{(\mathbf{x})}\}_{t=1}^T$ (and thus the one with the lowest regret), using the fact that the minimum regret is no higher than the average (with respect to $t \sim \mathrm{Uniform}(1, \dots, T)$), and noting that such an average is precisely $\frac{1}{T}$ times the cumulative regret.

B ADDITIONAL EXPERIMENTS AND DETAILS

We first provide additional experimental results in Section B.1. The additional experimental details (e.g., choice of kernel and description of baselines) are deferred to Section B.2.

B.1 Additional Experimental Results

Actions Sampled by Baselines. In this section, we first present the actions sampled by SafeOpt-MC and PredVar in Figure 3 (similar to Figure 1, which shows the actions selected by M-SafeOpt). While PredVar samples actions throughout the safe set in order to reduce uncertainty with respect to both f and g, SafeOpt-MC samples either close to the "safe boundary" (potential expanders) or among the potential maximizers. See Appendix B.1.1 for further discussion on the performance of these two algorithms, and how they compare with M-SafeOpt.

Effect of L_f and L'_g: In this section, we evaluate the performance of M-SafeOpt in the scenario that the parameters L_f and L'_g used in the algorithm differ from the corresponding true values. As mentioned in Section

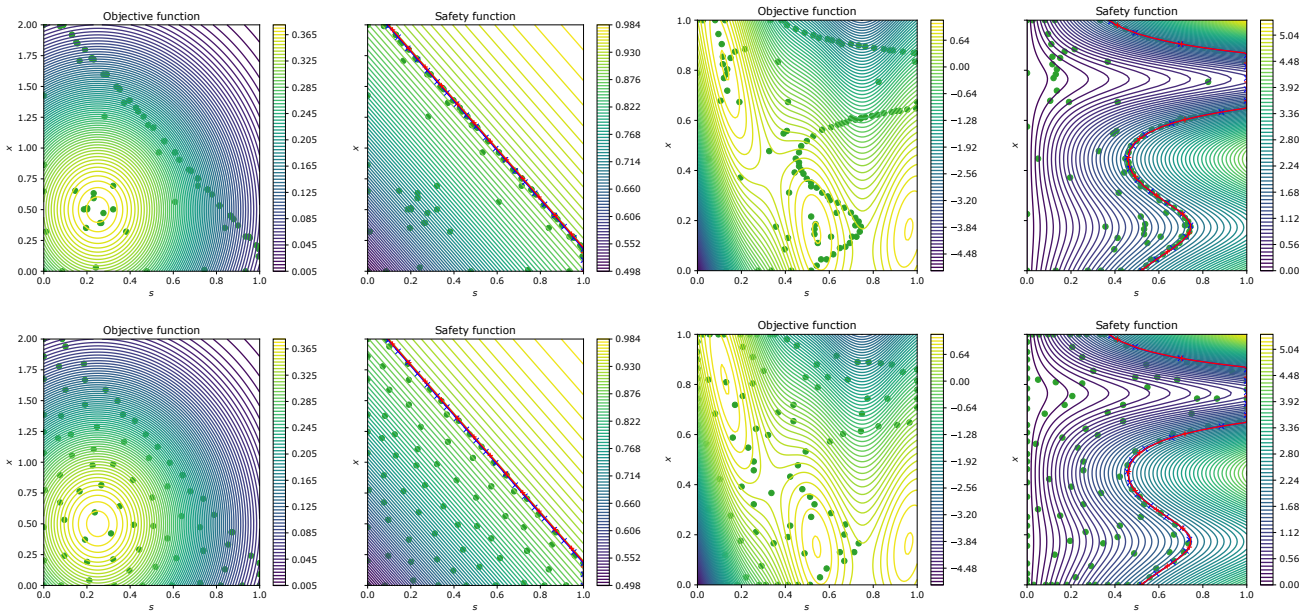

Figure 3: The first two columns shows the actions sampled by the SafeOpt (row 1) and PredVar (row 2) algorithms for the simulated clinical trial experiment, while the last two columns shows the corresponding plots for the synthetic 2D experiment (from Section 5). The true safe boundary is shown in red and the boundary discovered by the algorithm is shown in blue (in columns 2 and 4).

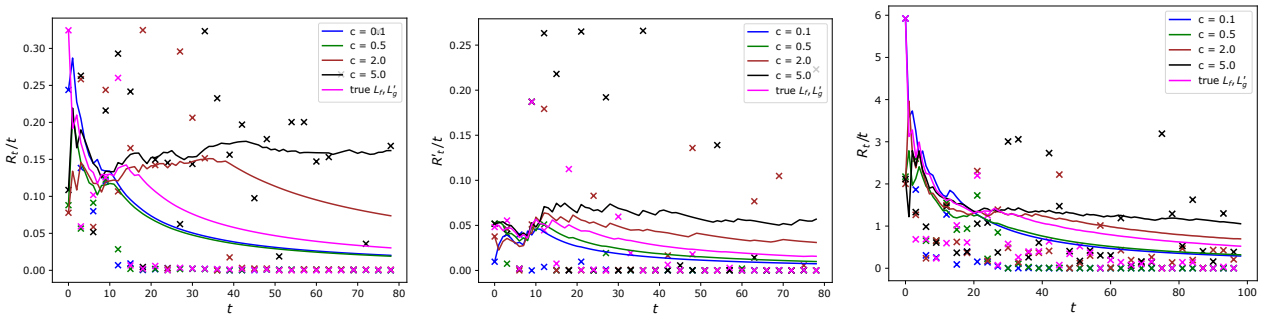

Figure 4: The first column shows the normalized cumulative regret, R_t/t, incurred by M-SafeOpt for the simulated clinical trial experiment for different values of c (where L_f is set to cL_f, and L'_g is set to L'_g/c), while the second column shows that for R'_t/t. The third column corresponds to the synthetic 2D experiment (showing R_t/t). The instantaneous regret values are shown using markers.

2, an overestimate of L_f and an underestimate of L'_g also allow our theoretical guarantees to hold (since (4) and (5) remain valid). Therefore, we design this set of experiments by varying the values of L_f and L'_g as follows: $L_f \leftarrow cL_f, L'_g \leftarrow L'_g/c$ with $c \in \{2, 5\}$. Additionally, to see how M-SafeOpt performs in the reverse scenario, i.e., L_f is underestimated and L'_g is overestimated, we also consider $c \in \{0.1, 0.5\}$.

Intuitively, $c > 1$ implies that the algorithm becomes more cautious when setting $\texttt{elim}_t = \texttt{true}$ and $\texttt{expd}_t = \texttt{false}$, and thus, the regret is expected to converge slower than that with $c = 1$. This is also observed in our experiments, as shown in Figure 4.

On the other hand, $c < 1$ leads to more aggressive elimination of $\mathbf{x}$'s and expansion of actions on the "safe boundary". This may cause the regret to converge faster (as is also the case in Figure 4). However, this may come at the expense of failing to explore potentially optimal regions due to improper elimination/non-expansion based on the incorrect estimates. In the present scenario with "well-behaved" objective and safety functions,

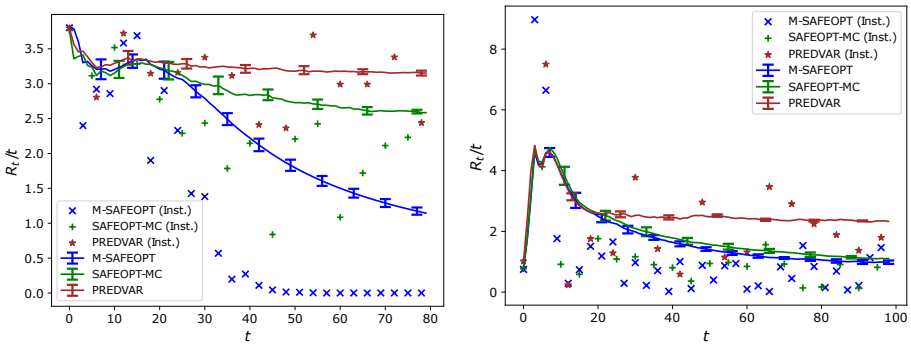

Figure 5: The first column shows the normalized cumulative regret, R_t/t, for the synthetic 3D experiment, while the second column shows that for the pendulum swing-up experiment. The corresponding instantaneous regret values are shown using markers.

this behavior tends to be avoided.

We also note that it is possible to be more robust to such choices of L_f and L'_g ($c < 1$), and also possibly imprecise knowledge of the kernel, by making the following change to the algorithm: eliminate $\mathbf{x}$'s from the whole set $\mathcal{D}_{\mathcal{X}}$ (instead of the previous set $\mathcal{D}^t_{\mathcal{X}}$) in each round, such that any $\mathbf{x}$ that may have been incorrectly eliminated in a specific round still has a chance to be recovered once the confidence bounds are refined in subsequent rounds. In view of this improved robustness, we adopt this strategy in our implementation.

In the rest of this subsection, we compare the performance of M-SafeOpt with the baseline algorithms on two other problems – one with a three dimensional input domain, and another being a modification of the pendulum swing-up problem, a classic control task (Brockman et al., 2016).

Synthetic 3D Functions. For this experiment, we used the Hartmann-3 function (Hedar, 2013) as the objective function $f_{\text{syn}_{3D}}$, along with the following safety function:

$$g_{\text{syn}_{3D}}(s, \mathbf{x}) = s + x_1^2 + x_2^3, \tag{118}$$

where $\mathbf{x} = (x_1, x_2)$. The domain is set to be $[0, 1]^3$, and is discretized into $75 \times 75 \times 75$ linearly spaced points. We set the safety threshold to $h = 2$, such that $g_{\text{syn}_{3D}}$ satisfies the assumptions of our problem setup, i.e., monotonicity with respect to s, and safety for $s = 0$ for all values of $\mathbf{x} \in \mathcal{D}_{\mathcal{X}}$. The function evaluations are set to be noiseless.

The results of running M-SafeOpt, along with the baseline algorithms, are presented in Figure 5. Similar to our previous experiments, we see that M-SafeOpt is able to converge to the global safe optimum while attaining sublinear regret, whereas the regret incurred by SafeOpt-MC and PredVar is not sublinear, since they continue to explore suboptimal regions.

Pendulum Swing-up Problem. For this problem, we consider the pendulum swing-up problem, which is a classic control problem available as an OpenAI Gym environment (Brockman et al., 2016). The task is to apply torque to the free end of a pendulum such that it swings and stays in the upright position, i.e., it attains an angular velocity of zero when it reaches this position. We adapt the problem to our setup as follows. The initial angle ($\mathbf{x}$) of the pendulum lies within the range $\mathcal{D}_{\mathcal{X}} = [-2\pi + \pi/36, -\pi - \pi/36]$ (with angle $= 0$ denoting the upright position), and the torque (s) lies in $\mathcal{D}_{\mathcal{S}} = [0, 1]$ (s is scaled up by a factor of 40 to calculate the motion, so that the upright position is attainable). The torque is applied only once at the beginning, while the episode is run for 100 time steps.

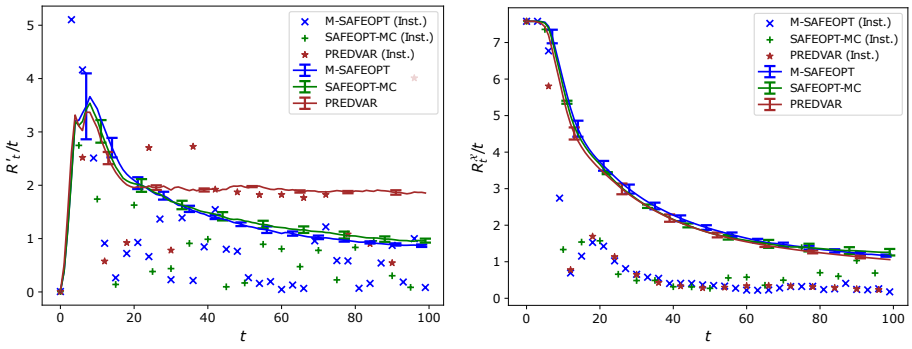

Figure 6: The first column shows the normalized cumulative regret, R'_t/t, for the pendulum swing-up experiment, while the second column shows that for $R^{\mathcal{X}}_t/t$ when running M-SafeOpt (Case 2) along with the baseline algorithms. The corresponding instantaneous regret values are shown using markers.

The reward function is defined as follows (similar to (Losalka and Scarlett, 2023)):

$$f_n(s,\mathbf{x}) = \begin{cases} -\theta_n^2(s,\mathbf{x}) - \frac{\dot{\theta}_n^2(s,\mathbf{x})}{10} - \frac{s^2}{1000}, & \text{if } \theta_n(s,\mathbf{x}) \le 0 \\ -\dot{\theta}_{up}(s,\mathbf{x}) & \text{if } \theta_n(s,\mathbf{x}) > 0, \end{cases} \tag{119}$$

$$f(s,\mathbf{x}) = \max_{n \le 100} f_n(s,\mathbf{x}), \tag{120}$$

where $\theta_n(s,\mathbf{x})$ and $\dot{\theta}_n(s,\mathbf{x})$ denote the angle and angular velocity of the pendulum at the n^{th} time step, and $\dot{\theta}_{up}(s,\mathbf{x})$ denotes the angular velocity of the pendulum when it crosses the upright position, starting with an initial angle and torque of $\mathbf{x}$ and s respectively.

The safety function is set to be equal to the maximum magnitude of the angular velocity attained by the pendulum at any time step during the episode:

$$g(s,\mathbf{x}) = \max_{n \le 100} |\dot{\theta}_n(s,\mathbf{x})|, \tag{121}$$

and the safety threshold is set to $h = 9$. In this experiment, we consider noisy queries for both f and g, namely, additive $\mathcal{N}(0, 0.05)$ noise.

The initial angular velocity is always set to zero such that our safety assumption is satisfied ($s = 0$ being safe for every $\mathbf{x}$); this is because the threshold magnitude of velocity is unattainable for any value of the initial angle ($\mathbf{x} \in \mathcal{D}_{\mathcal{X}}$), unless a positive torque ($s > 0$) is applied. One episode (with 100 time steps) is simulated to calculate the reward and safety values in every iteration of optimization.

The input domain is discretized into 100×100 linearly spaced points, and the results of running M-SafeOpt, along with the baseline algorithms are presented in Figure 5. In this case, M-SafeOpt and SafeOpt-MC behave similarly, whereas PredVar incurs higher regret due to its exploratory nature.

B.1.1 Further Discussion

With respect to the performance of the baseline algorithms SafeOpt-MC and PredVar , we highlight the following observations:

- SafeOpt-MC tries to simultaneously explore the potentially optimal regions (M_t) and actions that could potentially expand the safe set (G_t) as seen via the actions sampled in Figure 3. However, it expands while being "blind" to the objective function f (i.e., it does not try to evaluate the suboptimality of the potential expanders). This appears to be unavoidable in the general scenario, since without any known structure on g (e.g., monotonicity), it is not possible to predict which region of the input domain may contain the optimal safe action; hence, the entire reachable safe set must be explored by SafeOpt. This is reflected in the

performance of SafeOpt with respect to R_t and R'_t (e.g., see the first and second columns of Figure 2, and the corresponding plots in Figure 5 and 6).

- Due to its exploratory nature, PredVar performs well in terms of $R_t^{\mathcal{X}}$ (as observed in the third column in Figure 2), i.e., it is able to identify a near-optimal s for each $\mathbf{x} \in \mathcal{D}_{\mathcal{X}}$. However, for the same reason, it performs poorly with respect to R_t and R'_t, i.e., it does not make progressively "better" choices (since it tries to reduce uncertainty over f and g over S_t without considering potential optimality separately), as observed in the first and second columns of Figure 2, as well as both plots in Figure 5 and the first column of Figure 6.

B.2 Details of Experiments

In this section, we describe the details of the setup of our experiments, including the details of the objective and safety functions, and the implementation details of the algorithms.

B.2.1 Experimental Setup

Simulated Clinical Trial. For this experiment, we consider the dose efficacy (f_{eff}) and dose toxicity (g_{tox}) functions as described in (37) and (38) respectively. Specifically, the values of θ_i's are set as follows: $\theta_0^f = 1, \theta_1^f = 2, \theta_2^f = 1, \theta_3^f = -4, \theta_4^f = -1, \theta_1^g = 2, \theta_2^g = 1$ to give us the functions shown in Figure 1. As stated in Section 5, we assume that d_1 and d_2 represent the dosages of two different drugs that are administered as a combination in a clinical trial. For drug combinations, it is common to observe non-monotonic behavior of the efficacy (e.g., immunotherapy trials (Cai et al., 2014)). Since the toxicity increases monotonically with respect to both d_1 and d_2 in this experiment, either could be treated as the safety variable s. We use d_1 as the safety variable in our implementation, and try to find the global safe optimal dose combination for goal (i) (as in Section 2).

For goal (ii) (as in Section 2), the same functions are used in our experiment. Since we need to find the optimal safe d_1 for every d_2 in this case. From the perspective of practical relevance, it may be more sensible to interpret d_2 as the age of patients, so that the goal translates to finding the optimal dose (d_1) for every age (d_2). However, from the perspective of evaluation of the algorithms, the proposed meaning/interpretation of each variable does not have any quantitative effect.

The input domain is set to be $[0, 1] \times [0, 2]$, which is discretized into 200×200 linearly spaced points in the domain. The function evaluations observed by the algorithm are noiseless. The safety function $g_{\text{tox}}(s, \mathbf{x})$ satisfies the assumptions of our problem setup, i.e., strict monotonicity with respect to s, and safety at $s = 0$ for all $\mathbf{x} \in \mathcal{D}_{\mathcal{X}}$. However, f_{eff} is non-monotonic in both s and $\mathbf{x}$.

Synthetic 2D Functions. We consider the following synthetic functions:

$$f_{\text{syn}_1} = \alpha \cdot \left((x - bs^2 + cs - 6)^2 + 10(1 - t)\cos(s) + \Delta\right), \tag{122}$$

$$g_{\text{syn}_1} = 2s(e^y \sin(10y) + \sin(5y) + 5)/3, \tag{123}$$

where the parameters are set to the following values: $\alpha = 1/51.95, \Delta = -44.81, b = 5.1/4\pi^2, c = 5/\pi, t = 1/8\pi$ as per those used for defining the scaled Branin function (Picheny et al., 2013). y is set to $x + 1/3$ for defining g_{syn_1}, so that one of the three local optima of the objective function gets excluded from the safe region. We use these functions due to the presence of multiple local optima (of f_{syn_1}), less smooth optimization surfaces for both f_{syn_1} and g_{syn_1} (compared to f_{eff} and g_{tox}), and difficulty in eliminating $\mathbf{x}$'s and/or terminating expansion due to presence of near-optimal actions in the unsafe region close to safe boundary. This makes it more difficult for the algorithm to identify and eliminate suboptimal regions of the input space. The function evaluations observed by the algorithm are noiseless.

The input domain is set to be $[0, 1] \times [0, 1]$, and is discretized into 200×200 linearly spaced points in the domain. The function evaluations observed by the algorithm are noiseless. The safety function $g_{\text{syn}_1}(s, \mathbf{x})$ satisfies strict monotonicity with respect to s, and safety at $s = 0$ for all $\mathbf{x} \in \mathcal{D}_{\mathcal{X}}$.

B.2.2 Implementation Details

In all our experiments, β_f^t and β_g^t are set to a constant value of 3 in all rounds $t \geq 1$. The use of a constant β is fairly common in the Bayesian optimization/safe Bayesian optimization literature, since the theoretical values

tend to be overly cautious to aid the derivation of theoretical guarantees.

We use the Trieste library for Bayesian optimization in our implementations (Picheny et al., 2023). All experiments are repeated 5 times, and the plots in Figure 2 and 5 show the mean values of the corresponding notions of regret (both instantaneous and cumulative), along with the standard deviations via error bars.

Gaussian Process Model. In all our experiments, the Gaussian process models for both f and g use the Matérn-$\frac{5}{2}$ kernel, with the length scales and variance of the kernel set to be trainable. A log-normal prior is used for both the variance and the length scales with a standard deviation 1. The mean values for the length scales are set to 0.2, and that for the variance is set to 1 for the simulated clinical trial and the synthetic 3D experiments, and 9 for the experiment with the synthetic 2D functions. The GP models assumes a low noise level of 10^{-5} for numerical stability (except for the pendulum swing-up experiment, where the algorithm observes noisy evaluations of f and g, the noise being sampled from $\mathcal{N}(0, 0.05)$ and the GP model assumes a noise level with variance 0.05). Seeking minimal manual tuning, the choices of parameters mostly follow the recommendations in (Picheny et al., 2023).

M-SafeOpt Algorithm. For implementing our M-SafeOpt algorithm, we first find the values of L_f and L'_g by computing the gradients of f and g over a finely discretized grid of points in the input space. We directly use these values unless stated otherwise, but we recall that Appendix B.1 also explores the algorithm's robustness to misspecified values.

We also note that while our theory holds for continuous $\mathbf{x}$, our implementation relies on discretization of $\mathbf{x}$ due to the explicit for-loop over $\mathbf{x}$. For continuous domains, alternative approaches involving approximations are possible; however, we do not claim the validity of our theoretical guarantees for such variations. A few such alternatives are discussed in Appendix C.2 in (Losalka and Scarlett, 2023) for the case that $f = g$, and similar approaches can also be adopted in our setting with distinct f and g.

SafeOpt Algorithm. For implementing the SafeOpt algorithm (Sui et al., 2015), we specifically use the variant proposed by Berkenkamp et al. (2021), SafeOpt-MC, that works with distinct objective and safety functions. To incorporate the knowledge of monotonicity of g in the implementation, we explicitly define the set G_t of *potential expanders* as the set of actions on the current "safe boundary" as discovered by the algorithm and as defined by M-SafeOpt. The only difference is that in this case, we remove any action with $s_t^{(\mathbf{x})} = 1$ from G_t; this is because SafeOpt defines G_t as the set of actions that could potentially expand the current safe set of actions, whereas $s_t^{(\mathbf{x})} = 1$ implies that $g(s, \mathbf{x})$ has been discovered to be safe for every $s \in \mathcal{D}_{\mathcal{S}}$ already. On the other hand, the set of *potential maximizers*, M_t, is computed exactly as defined in (Sui et al., 2015). We avoid using Lipschitz constants, instead relying on the modification proposed by (Berkenkamp et al., 2017).

PredVar Algorithm. For the PredVar algorithm, we conceptually rely on (Schreiter et al., 2015), while extending the algorithm to consider multiple functions simultaneously. We define the the currently known safe region in the same way as that in M-SafeOpt. The acquisition function uses the maximum among the width of the confidence intervals given by the GP models for f and g for all actions in the safe set S_t. Thus, PredVar behaves as a purely exploratory algorithm that tries to minimize variance across the safe input domain for both f and g simultaneously, while also expanding the safe set.

C DISCUSSION

C.1 Necessity of L_f and L'_g

Recall that we assume a minimum growth rate of g with respect to s (i.e., $L'_g > 0$) and a a maximum growth rate of f with respect to s (i.e., $L_f < \infty$). Here we argue that these assumptions (or similar) are in fact essential for attaining meaningful regret bounds in our setting.

To see this, we consider the highly simplified special case in which there is only a *single* choice of $\mathbf{x}$ (i.e., $|\mathcal{D}_{\mathcal{X}}| = 1$), and the goal is to maximize f with respect to $s \in [0, 1]$ alone, subject to safety. When $L'_g = 0$, we can encounter a situation such as that shown on Figure 7, where g is flat with a value extremely close to h, say $h - \epsilon$. For arbitrarily small ϵ, it is arbitrarily hard to identify (to within a constant accuracy, say ± 0.01) the value of s at

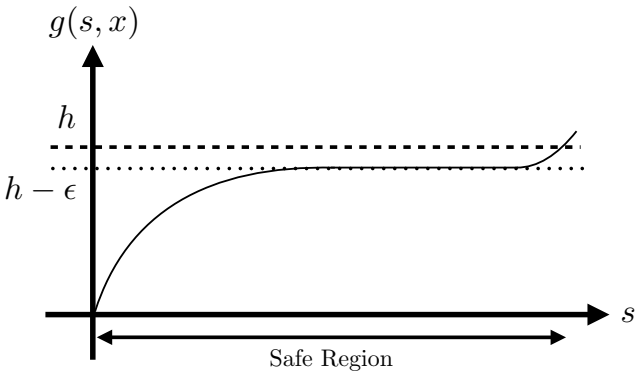

Figure 7: Example of a function where identifying the safe boundary (i.e., the highest safe s) can be arbitrarily hard.

which the function crosses from safe to unsafe.[3] On the other hand, if f is *not* flat (i.e, $L_f > 0$), then accurately identifying that crossing point is crucial for optimizing f.

Along similar lines, even if we have $L'_g > 0$, having $L_f = \infty$ would imply that even a minuscule amount of inaccuracy with respect to identifying the safe boundary (which is almost always unavoidable, particularly in the noisy setting) can lead to strict suboptimality with respect to f due to abrupt changes.

Thus, the assumptions $L'_g > 0$ and $L_f < \infty$, or possibly similar kinds of assumptions with different specific details, are essential for the goals of our paper. We note that while the assumption $L'_g > 0$ is somewhat specific to our setting, growth rate upper bounds (i.e., Lipschitz constants) are much more common, e.g., as used by existing algorithms such as SafeOpt. Moreover, for many commonly-considered kernels (e.g., Matérn with $\nu > 1$), Lipschitz continuity with respect to s *and* $\mathbf{x}$ is automatically guaranteed for any function in the RKHS (e.g., see Remark 5 of Shekhar and Javidi (2020)).

As a side note, we point out that strict monotonicity of $g(\cdot, \mathbf{x})$ alone may not directly imply $L'_g > 0$, either due to the presence of a global minimum at $g(0, \mathbf{x})$, or due to inflection points. For example, $g(s, \mathbf{x}) = s^2$ is strictly monotonically increasing in s, but the minimum value of the partial derivative with respect to s is 0. While strict monotonicity still implies (5) for some $L'_g > 0$ when $s > s'$, the difference is that L'_g needs to vary with s' (and possibly approach 0 as s' approaches s).

C.2 Variations and Extensions

In our paper, we have modeled f and g separately for clarity of exposition. However, we can easily handle joint modeling (e.g., to capture correlations between f and g) in the same way as existing works such as (Berkenkamp et al., 2021): We simply define $h(\cdot, \cdot, 1) = f(\cdot, \cdot)$ and $h(\cdot, \cdot, 2) = g(\cdot, \cdot)$, and assume that the "expanded" function $h(s, \mathbf{x}, i)$ has a low RKHS norm (instead of f and g separately). The existing confidence bounds can then simply be applied to h, rather than to f and g separately.

In certain applications, the current problem setup may need to be extended to consider context variables c_t that cannot be "selected", but are rather provided by the environment in every round (e.g., a patient's blood sugar level in the adaptive clinical trial application). Furthermore, safety may be dictated by multiple safety functions $g_1, g_2, \ldots, g_l$, instead of a single function g. In both these scenarios, we note that our algorithms can be readily extended following the ideas proposed in (Berkenkamp et al., 2021), assuming each g_i satisfies our monotonicity assumption.

[3] For instance, supposing additive Gaussian noise, it is well-known that $\Theta\big(\frac{1}{\epsilon^2}\big)$ queries are needed to distinguish between function values of $h + \epsilon$ and $h - \epsilon$, and similarly for other values in between the two.