Benefits of Monotonicity in Safe Exploration with Gaussian Processes

Arpan Losalka[1] **Jonathan Scarlett**[1,2,3]

[1]Department of Computer Science, National University of Singapore, Singapore
[2]Department of Mathematics, National University of Singapore, Singapore
[3]Institute of Data Science, National University of Singapore, Singapore

Abstract

We consider the problem of sequentially maximising an unknown function over a set of actions while ensuring that every sampled point has a function value below a given safety threshold. We model the function using kernel-based and Gaussian process methods, while differing from previous works in our assumption that the function is monotonically increasing with respect to a *safety variable*. This assumption is motivated by various practical applications such as adaptive clinical trial design and robotics. Taking inspiration from the GP-UCB and SafeOpt algorithms, we propose an algorithm, monotone safe UCB (M-SafeUCB) for this task. We show that M-SafeUCB enjoys theoretical guarantees in terms of safety, a suitably-defined regret notion, and approximately finding the entire safe boundary. In addition, we illustrate that the monotonicity assumption yields significant benefits in terms of the guarantees obtained, as well as algorithmic simplicity and efficiency. We support our theoretical findings by performing empirical evaluations on a variety of functions, including a simulated clinical trial experiment.

1 INTRODUCTION

The sequential optimisation of an unknown and expensive-to-evaluate function f is a fundamental task with a number of interesting challenges. This task arises in various real-world applications, such as robotics [Lizotte et al., 2007], hyperparameter tuning in machine learning [Snoek et al., 2012], environmental monitoring [Srinivas et al., 2012], adaptive clinical trial design [Takahashi and Suzuki, 2021b], recommendation systems [Vanchinathan et al., 2014], and many others. Gaussian process (GP) based techniques such as GP-UCB [Srinivas et al., 2012], Thompson Sampling [Thompson, 1933] and Expected Improvement [Mockus et al., 1978] are particularly popular for this task.

In recent years, a variety of works have considered the important issue of *safety*, where some actions (function inputs) need to be avoided altogether. Various algorithms such as SafeOpt [Sui et al., 2015], StageOpt [Sui et al., 2018] and SafeOpt-MC [Berkenkamp et al., 2021] have been proposed to tackle this problem. The main idea behind these algorithms is to start with a safe seed set of inputs, and sequentially expand and explore the candidate set of potentially safe points to eventually identify a *reachable safe set* and/or the maximiser within that set.

In this work, our main goal is to show that *monotonicity with respect to just a single input variable* can be highly beneficial for this task. Consider a function f that we would like to maximise while ensuring that all selected points have value at most h.[1] We assume that the unknown function f is monotonically increasing (not necessarily strictly increasing) with respect to a *safety variable* $s \in [0, 1]$, while possibly remaining highly non-monotone with respect to the remaining variables $\mathbf{x}$. We consider performance measures based on both a form of cumulative regret and a notion of identifying the entire safe region. Briefly, the benefits of monotonicity in s are:

(i) Our theoretical bounds have improved dependencies over previous works, particularly with respect to the domain size (see Appendix C.1 for details).
(ii) By exploiting the monotonicity, we can circumvent the need to explicitly keep track of potential expanders, as existing algorithms do.
(iii) Under monotonicity, continuity, and the mild additional assumption that $s = 0$ is always safe, we show that every safe point is reachable, which is not the case for general non-monotone functions.

Intuitively, the presence of the safety variable allows the

[1]This is distinct from previous works that require a value of *at least* h, and we discuss the differences in Section 2.

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

algorithm to choose how cautious it should be while exploring the domain, and creates a more favourable function landscape for exploration. For instance, the algorithm can "back off" or "proceed with caution" (lower s) when considering less-explored $\mathbf{x}$ values, but subsequently act more aggressively (higher s) when it becomes more confident that it is safe to do so.

Applications: Consider the task of adaptive clinical trial design, where the goal is to recruit patients for drug trials in order to evaluate the safety and efficacy of a drug or drug combinations [Coffey and Kairalla, 2008, Berry, 2006]. It is well-accepted that patient characteristics play a significant role in both safety and efficacy given a drug dose [Lee et al., 2020]. Thus, it is helpful to model the drug dose as the safety variable s, and patient characteristics as the variable $\mathbf{x}$, both of which influence the unknown toxicity, say f_{tox}. For many classes of drugs such as cytotoxic agents, the toxicity and efficacy both increase strictly as the drug dose is increased [Chevret, 2006]. In such cases, for Phase I clinical trials, it is usually necessary to find the *Maximum Tolerated Dose (MTD)*, which is the dose with the maximum toxicity within the permitted threshold for the patient characteristics under consideration [Aziz et al., 2021, Riviere et al., 2014, Shen et al., 2020]. Formulating this task in our problem setting can result in (i) maximisation of benefits (via regret minimisation) and (ii) minimisation of harmful effects to patients involved in the study (via safety constraints), while simultaneously (iii) identifying safety information about the entire set of patient characteristics (via sub-level set estimation), each of which are important goals of adaptive clinical trial design. GP optimisation has been recently used for this task [Takahashi and Suzuki, 2021b,a], but the safety constraints were met in these works by being highly cautious in dose increments (single step increments for discrete dosage levels), and no theoretical guarantees were sought.

Problems in robotics may also serve as potential applications for our problem setup. For example, consider the scenario where a robot performs a task with certain parameters given by the variable $\mathbf{x}$, but that there also exists a parameter s indicating the *speed* (or more generally, any measure of "caution" with lower values being more cautious) at which the task is attempted. Then, one may seek to optimise the parameters while ensuring that s is never pushed too high to become unsafe, leading to a natural monotonicity constraint. We explore a simple inverted pendulum problem of this kind in Section 5. We note that in some cases, it may be more natural to have separate functions f and g for measuring reward and safety, and we discuss such variations in Section 3.

Related Work: Sui et al. [2015] proposed the first algorithm, SafeOpt, for safe GP optimisation. StageOpt was proposed by Sui et al. [2015] as a variation of SafeOpt, where safe set finding and function optimisation were separated into two distinct phases. Berkenkamp et al. [2021] proposed a generalised version of SafeOpt called SafeOpt-MC to tackle the problem when safety functions are decoupled from the function being optimised. Other algorithms such as GOOSE [Turchetta et al., 2019] seek to be more goal-directed during safe set expansion. Safe exploration using Gaussian processes has also been considered by Schreiter et al. [2015], but for the goal of active learning of the unknown function. To our knowledge, none of these works have explored the benefits of having a safety variable leading to monotonicity. Moreover, their theoretical guarantees exhibit certain weaknesses with respect to the domain size that we are able to circumvent; see Section 4 for the details.

Another related line of work introduces algorithms such as GoSafe [Baumann et al., 2021] and GoSafeOpt [Sukhija et al., 2022] to address safe Bayesian optimisation problems. Their setting is fundamentally different to ours, since they consider a *dynamic* system with time-varying inputs, and allow *intervening* with a safe backup policy when an imminent safety violation is detected in the original policy. Our setup considers static functions without interventions (which may not be feasible in some applications, e.g., once a drug dose is administered it may not be possible to change).

A different approach to safe GP optimisation is taken in [Amani et al., 2021], in which conditions are explored under which an initial safe seed set can be sampled enough times for the resulting samples alone to expand the safe set significantly (and include the global safe maximiser). However, this requires careful assumptions on the seed set depending strongly on the kernel, and the idea appears to be most suited to finite-dimensional feature spaces (e.g., linear or polynomial kernels); see Appendix C.3 for discussion.

In a parallel line of work, the problem of level set estimation and related settings involving excursion sets has been considered, e.g., see [Gotovos et al., 2013, Bogunovic et al., 2016, Bolin and Lindgren, 2015]. GP optimisation with monotonicity assumptions has also been considered by Li et al. [2017] and Wang and Welch [2018]. However, these works do not consider safety constraints, and consequently, the associated algorithms are significantly different.

A notable prior work combining safety and monotonicity is [Wang et al., 2022], but they study a non-GP setting where all the arms (corresponding to $\mathbf{x}$ in our setting) are modeled separately, and the goal is best-arm identification. This leads to a precise characterisation of the number of arm pulls. However, their setup, algorithm, and results remain very different from our work, where smoothness with respect to $\mathbf{x}$ (as well as s) plays a crucial role.

Finally, safety has been considered in a variety of other settings including linear bandits [Amani et al., 2019, Khezeli and Bitar, 2020] and reinforcement learning [Turchetta et al., 2016, Berkenkamp et al., 2017, Turchetta et al., 2020], but

compared to the works outlined above for GP settings, these are less directly relevant to ours.

Contributions: Summarising the above discussions, we list our main contributions as follows:

- We study the problem of safe sequential optimisation of an unknown function, and introduce the idea of considering monotonicity of the function with respect to a "safety" variable. We propose the monotone safe UCB (M-SAFEUCB) algorithm for this problem.
- We show that with high probability, M-SAFEUCB achieves sub-linear regret (for a suitably-defined regret notion to follow), only selects safe actions, and identifies the safe (sub-level) set of the function with high accuracy.
- We experimentally evaluate M-SAFEUCB alongside other baselines on a variety of functions, and demonstrate that the resulting performance aligns with the theoretical guarantees.

2 PROBLEM STATEMENT

We consider the problem of sequentially maximising a fixed but unknown function $f : \mathcal{D} \to \mathbb{R}$ over a set of decisions while satisfying safety constraints, where $\mathcal{D} = \mathcal{D}_{\mathcal{S}} \times \mathcal{D}_{\mathcal{X}}$, $\mathcal{D}_{\mathcal{X}} \subset \mathbb{R}^d$ is a compact set and $\mathcal{D}_{\mathcal{S}} = [0, 1]$. As discussed above, we assume that the function is monotonically increasing in the first argument. At each round t, an algorithm selects an *action* $(s_t, \mathbf{x}_t) \in \mathcal{D}_{\mathcal{S}} \times \mathcal{D}_{\mathcal{X}}$, and subsequently observes the *noisy reward* $y_t = f(s_t, \mathbf{x}_t) + \epsilon_t$. The action must be chosen at round t such that it depends upon the actions picked and the rewards observed up to round $t-1$, denoted by $\mathcal{H}_{t-1} = \{(s_k, \mathbf{x}_k, y_k) : k = 1, \ldots, t-1\}$ (i.e., the history). The algorithm is also required to satisfy the safety constraint $f(s_t, \mathbf{x}_t) \leq h \ \forall t \geq 1$ with high probability.

Goal: The goals of an algorithm in our problem setting include maximising its cumulative reward and/or finding the entire h-sub-level set of f, while only choosing safe actions. These desiderata are formalised as follows. For cumulative regret, we consider the following definition:

$$R_T = \sum_{t=1}^{T} r_t, \text{ where } r_t = h - f(s_t, \mathbf{x}_t), \tag{1}$$

where we compare against h rather than $\max_{s,\mathbf{x}\in\mathcal{D}} f(s, \mathbf{x})$ in view of the safety requirement. For the sub-level set, we define

$$L_h(f) = \{(s, \mathbf{x}) \in \mathcal{D} | f(s, \mathbf{x}) \leq h\}, \tag{2}$$

which we seek to approximate to high accuracy (see below). For the safety requirement, we seek that the sampled points satisfy $f(s_t, \mathbf{x}_t) \leq h \ \forall t \geq 1$ with high probability.

Returning to the notion of the safe sub-level set, we quantify the quality of a solution $\hat{L}$ returned by an algorithm after T rounds, with respect to a given point $(s, \mathbf{x}) \in \mathcal{D}$, using the following misclassification loss:

$$l_h(s, \mathbf{x}) = \begin{cases} \max\{0, \ \ h - f(s, \mathbf{x})\} & \text{if } (s, \mathbf{x}) \notin \hat{L}, \\ \infty, & \text{if } (s, \mathbf{x}) \in \hat{L} \text{ and } (s, \mathbf{x}) \notin L_h(f), \\ 0, & \text{if } (s, \mathbf{x}) \in \hat{L} \text{ and } (s, \mathbf{x}) \in L_h(f). \end{cases} \tag{3}$$

Essentially, this loss function penalises an algorithm heavily for classifying unsafe points as safe, while the penalty for classifying safe points as unsafe increases linearly with the difference in the function value from the threshold. We require that the algorithm should return an *ϵ-accurate solution* with probability at least $1 - \delta$, i.e.,

$$\mathbb{P}\left\{\max_{s,\mathbf{x}\in\mathcal{D}} l_h(s, \mathbf{x}) \leq \epsilon\right\} \geq 1 - \delta. \tag{4}$$

We note that the notion of regret that we consider is primarily of interest *when coupled with* (4), rather than in itself. Small R_T is generally desirable since it implies that we are eventually sampling points with the highest possible safe function value. However, one way of achieving small R_T might be to always choose the same $\mathbf{x}$ and gradually increasing s until a low-regret point is found. The additional condition (4) precludes this possibility.

Assumptions: Certain smoothness assumptions on the function f are necessary in order to be able to provide theoretical guarantees. Similar to much of the earlier work in the area of GP optimisation, we assume that f has bounded norm in the reproducing kernel Hilbert space (RKHS) of functions $\mathcal{D} \to \mathbb{R}$, with positive semi-definite kernel function $k : \mathcal{D} \times \mathcal{D} \to \mathbb{R}$. This RKHS, denoted by $\mathcal{H}_k(\mathcal{D})$, is completely specified by its kernel function $k(\cdot, \cdot)$ and vice-versa, with an inner product $\langle \cdot, \cdot \rangle_k$ obeying the reproducing property: $f(\mathbf{z}) = \langle f, k(\mathbf{z}, \cdot) \rangle_k \ \forall f \in \mathcal{H}_k(\mathcal{D})$. The RKHS norm $||f||_k = \sqrt{\langle f, f \rangle_k}$ is a measure of the smoothness of f with respect to the kernel function k, and satisfies $f \in \mathcal{H}_k(\mathcal{D})$ if and only if $||f||_k < \infty$. We assume a known upper bound B on the RKHS norm of the unknown target function, i.e., $||f||_k \leq B$. We also adopt the standard assumption of bounded variance: $k(\mathbf{z}, \mathbf{z}) \leq 1 \ \forall \mathbf{z} \in \mathcal{D}$.

In addition, 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 (our algorithms are written for the discrete case, and we discuss the distinction between the two in Appendix C.2);
2. The function f is monotonically increasing in the first argument, i.e., for all $\mathbf{x} \in \mathcal{D}_{\mathcal{X}}$, $f(s, \mathbf{x})$ is a non-decreasing 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 $\mathbf{x} \in \mathcal{D}_{\mathcal{X}}$, $f(0, \mathbf{x}) \leq h$;
4. The function f exceeds the threshold h for at least one point in the domain, i.e., $\max_{(s,\mathbf{x}\in\mathcal{D})} f(s, \mathbf{x}) > h$.

The third assumption above is natural since $s = 0$ corresponds to the most cautious selection possible, and the fourth assumption is mild since otherwise every action is safe and hence no algorithm would ever choose an unsafe action.

Finally, the noise sequence $\{\epsilon_t\}_{t\geq 1}$ is assumed to be conditionally R-sub-Gaussian for a fixed constant $R \geq 0$, i.e.,

$$\forall t \geq 0, \forall \lambda \in \mathbb{R}, \mathbb{E}\left[e^{\lambda \epsilon_t} | \mathcal{F}_{t-1}\right] \leq \exp\left(\frac{\lambda^2 R^2}{2}\right) \quad (5)$$

where $\mathcal{F}_{t-1}$ is the σ-algebra generated by the random variables $\{s_k, \mathbf{x}_k, \epsilon_k\}_{k=1}^{t-1}$ and $\mathbf{x}_t$.

Difference to Existing Settings: An important distinction between our work and certain previous ones (e.g., [Sui et al., 2015, 2018]) is that we consider all points *below* the threshold to be safe, rather than all points above the threshold. Our setting corresponds to trying to maximise a function while avoiding the risk of pushing it too far (e.g., dosage in clinical trials), whereas the alternative setting corresponds to needing to avoid excessively low-performance decisions (e.g., parameter configurations that may cause a drone to crash). The resulting algorithms are somewhat different since in our setting, maximisation algorithms will have a natural tendency to move closer to the safety threshold.

While the above difference is important to keep in mind, it is also worth noting that it becomes insignificant when one considers variations of the problem with *separate* safety and reward functions (see Section 3 for further discussion). In addition, in our experiments we adapt SafeOpt to suit our setting.

3 PROPOSED ALGORITHM

Gaussian Process Model: As is common in prior works, we consider algorithms that use Bayesian modeling (despite the non-Bayesian problem formulation). For this purpose, we use a Gaussian likelihood model for the observations, and a Gaussian process (GP) prior for uncertainty over the unknown function f. We let $GP(\mu(\cdot), k(\cdot, \cdot))$ denote a GP with mean μ and kernel k. In the following, we often shorten the GP input $(s, \mathbf{x})$ to $\mathbf{z}$ to simplify notation.

The algorithm uses a zero-mean GP, $GP(0, k(\cdot, \cdot))$, with k being the same as that defining the RKHS. The Gaussian likelihood has an associated variance parameter, which we denote by λ (i.e., corresponding to additive $\mathcal{N}(0, \lambda)$ noise in the Bayesian model).

With the Bayesian model in place, we have the following standard posterior update equations:

$$\mu_t(\mathbf{z}) = k_t(\mathbf{z})^T \left(K_t + \lambda I\right)^{-1} \mathbf{y}_t, \quad (6)$$

$$k_t\left(\mathbf{z}, \mathbf{z}'\right) = k\left(\mathbf{z}, \mathbf{z}'\right) - k_t(\mathbf{z})^T \left(K_t + \lambda I\right)^{-1} k_t\left(\mathbf{z}'\right), \quad (7)$$

$$\sigma_t^2(\mathbf{z}) = k_t(\mathbf{z}, \mathbf{z}). \quad (8)$$

Proposed Algorithm: We propose an algorithm called monotone safe UCB (M-SafeUCB), and provide its theoretical guarantees in Section 4. The key idea is to exploit the knowledge that the function f is monotonically increasing in the first argument, s, and thus, continually sample points in the domain that have their upper confidence bound equal to the threshold value h.

In more detail, M-SafeUCB uses a (standard) combination of the current posterior mean and standard deviation to construct an upper confidence bound (UCB) envelope for the function f over $\mathcal{D}$, given by

$$\mathrm{UCB}_{t-1}(s, \mathbf{x}) = \mu_{t-1}(s, \mathbf{x}) + \beta_t \sigma_{t-1}(s, \mathbf{x}), \quad (9)$$

where β_t is a time-dependent constant, that is set as per Lemma 1 below. In each round t, it chooses a sample such that $\mathrm{UCB}_{t-1}(s, \mathbf{x}) = h$ (favouring higher s in the rare case of having multiple such s for a single $\mathbf{x}$). This trades off between exploration and exploitation, i.e., it leads to selection of more points close to the currently optimal solution while exploring as well. If multiple such points are available with UCB equal to h, then M-SafeUCB selects the one that has the maximum posterior variance, thus helping to reduce uncertainty and encourage exploration.

To account for all possibilities that may arise in each round t, we set the candidate $s_t^{(\mathbf{x})} \in \mathcal{D}_{\mathcal{S}}$ and the candidate set $S_t^{(\mathbf{x})}$ for each $\mathbf{x} \in \mathcal{D}_{\mathcal{X}}$ as follows:

- If there exists $(s, \mathbf{x}) \in \mathcal{D}$ such that $\mathrm{UCB}_{t-1}(s, \mathbf{x}) = h$, then $s_t^{(\mathbf{x})} = \max\{s : (s, \mathbf{x}) \in \mathcal{D}, \mathrm{UCB}_{t-1}(s, \mathbf{x}) = h\}$ and $S_t^{(\mathbf{x})} = \{(s_t^{(\mathbf{x})}, \mathbf{x})\}$;
- If $\forall s \in \mathcal{D}_{\mathcal{S}}$, $\mathrm{UCB}_{t-1}(s, \mathbf{x}) > h$, then $s_t^{(\mathbf{x})} = 0$ (based on the assumption that for all $\mathbf{x} \in \mathcal{D}_{\mathcal{X}}$, $f(0, \mathbf{x}) \leq h$) and $S_t^{(\mathbf{x})} = \{(s_t^{(\mathbf{x})}, \mathbf{x})\}$;
- If $\forall s \in \mathcal{D}_{\mathcal{S}}$, $\mathrm{UCB}_{t-1}(s, \mathbf{x}) < h$, then $S_t^{(\mathbf{x})} = \emptyset$.

Next, the set S_t is formed by taking the union over all candidate sets $S_t^{(\mathbf{x})}$, i.e., $S_t = \bigcup_{\mathbf{x}\in\mathcal{D}_{\mathcal{X}}} S_t^{(\mathbf{x})}$. Then, $(s_t, \mathbf{x}_t)$ is chosen by maximising the predictive variance:

$$(s_t, \mathbf{x_t}) = \underset{(s,\mathbf{x})\in S_t}{\arg\max}\, \sigma_{t-1}(s, \mathbf{x}). \quad (10)$$

We note that if no $(s, \mathbf{x}) \in \mathcal{D}$ satisfies $\mu_{t-1}(s, \mathbf{x}) + \beta_t \sigma_{t-1}(s, \mathbf{x}) \geq h$ for a certain $t \leq T$, then it must be the case that with high probability, the entire function f lies below the safety threshold. This is precluded by assumption

Algorithm 1 M-SAFEUCB

Input: Prior $GP(0, k)$, parameters R, B, λ, δ
for $t = 1, \ldots, T$ **do**
  $\beta_t = B + R\sqrt{2\left(\gamma_{t-1} + 1 + \ln(1/\delta)\right)}$
  $S_t = \emptyset$
  for $\mathbf{x} \in \mathcal{D}_{\mathcal{X}}$ **do** ▷ find max. safe $s\,\forall\mathbf{x}$
    if $\mathrm{UCB}_{t-1}(s, \mathbf{x}) > h\,\forall s : (s, \mathbf{x}) \in \mathcal{D}$ **then**
      $s_t^{(\mathbf{x})} = 0$
      $S_t = S_t \cup \{(s_t^{(\mathbf{x})}, \mathbf{x})\}$
    else if $\exists s : (s, \mathbf{x}) \in \mathcal{D}$, $\mathrm{UCB}_{t-1}(s, \mathbf{x}) = h$ **then**
      $s_t^{(\mathbf{x})} = \max\{s : (s,\mathbf{x}) \in \mathcal{D},$
        $\mathrm{UCB}_{t-1}(s, \mathbf{x}) = h\}$
      $S_t = S_t \cup \{(s_t^{(\mathbf{x})}, \mathbf{x})\}$
    end if
  end for
  if $S_t = \emptyset$ **then** ▷ if safe everywhere, set $s = 1\,\forall\mathbf{x}$
    for $\mathbf{x} \in \mathcal{D}_{\mathcal{X}}$ **do**
      $S_t = S_t \cup \{(1, \mathbf{x})\}$
    end for
  end if
  $(s_t, \mathbf{x}_t) = \arg\max_{s,\mathbf{x} \in S_t} \sigma_{t-1}(s, \mathbf{x})$
  Update posterior to get μ_t, σ_t
end for
$\overline{\mathrm{UCB}}_T(s, \mathbf{x}) = \min_{1 \le t \le T} \mathrm{UCB}_t(s, \mathbf{x})\,\forall(s, \mathbf{x}) \in \mathcal{D}$
for $\mathbf{x} \in \mathcal{D}_{\mathcal{X}}$ **do** ▷ form safe boundary
  if $\overline{\mathrm{UCB}}_T(s, \mathbf{x}) > h\,\forall s : (s, \mathbf{x}) \in \mathcal{D}$ **then**
    $\overline{s}_T^{(\mathbf{x})} = 0$
  else
    $\overline{s}_T^{(\mathbf{x})} = \max\{s : (s, \mathbf{x}) \in \mathcal{D}, \quad \overline{\mathrm{UCB}}_T(s, \mathbf{x}) \le h\}$
  end if
end for
$\hat{L}_T = \{(s, \mathbf{x}) \in \mathcal{D} : s \le \overline{s}_T^{(\mathbf{x})}\}$
return $\hat{L}_T$

4 of our problem statement. However, with a low probability, it may also be the case that the noisy observations make the function "appear" to be below the threshold h to the algorithm. In this low probability scenario, $\forall\mathbf{x} \in \mathcal{D}_{\mathcal{X}}$, we set $s_t^{(\mathbf{x})} = 1$ and select $(s_t, \mathbf{x}_t) \in S_t$ as earlier by maximising variance.

While we make the mild assumption that at least one unsafe point exists, if it happens that f is safe for all $s, \mathbf{x} \in \mathcal{D}$, the algorithm will still eventually identify $s = 1$ as safe for every $\mathbf{x}$. In this scenario, the problem essentially becomes an unconstrained optimisation problem, and our notion of regret is no longer appropriate (since every point has a strict gap to h).

Finally, at the end of T rounds, the algorithm considers the intersection of the confidence regions across all rounds and all $\mathbf{x} \in \mathcal{D}_{\mathcal{X}}$, and forms an estimate of the safe sub-level set with respect to the upper bound of this intersection (denoted by $\overline{\mathrm{UCB}}_T(s, \mathbf{x})\}$) as follows:

$$\overline{s}_T^{(\mathbf{x})} = \max\{s : (s, \mathbf{x}) \in \mathcal{D}, \overline{\mathrm{UCB}}_T(s, \mathbf{x}) \le h\},\ \forall\mathbf{x} \in \mathcal{D}_{\mathcal{X}}, \tag{11}$$

$$\hat{L}_T = \{(s, \mathbf{x}) \in \mathcal{D} : s \le \overline{s}_T^{(\mathbf{x})}\}. \tag{12}$$

Note on Two-Function Settings: Throughout the paper, we focus on the case that the function f dictates both the objective (higher is better) and the safety (too high is unsafe). However, our ideas can also be applied in scenarios where these functions differ; say, with $f(s, \mathbf{x})$ being the objective and $g(s, \mathbf{x})$ dictating the safety. In the following, suppose that whenever we query a point, we observe noisy samples from both f and g. The two functions both have RKHS norm at most B, and the algorithm can form two separate GP posteriors for them.

First suppose that both f and g are monotone with respect to s. Moreover, similar to the current setup, suppose that the objective is to find the highest possible s associated with each $\mathbf{x}$. Then, one can simply apply our main algorithm to g, with Theorem 2 guaranteeing that we (approximately) find the entire safe boundary. Since both f and g are monotone, the highest safe s for g is also the highest safe s for f, and the overall problem is essentially unchanged compared to the single-function setting. As an example, this scenario corresponds to the task of finding the *MTD* in clinical trials as discussed in section 1.

In general, one may be interested in scenarios where only g is monotone with respect to s, whereas f is more general. Even in such cases, Algorithm 1 could be used as an initial step to find the safe boundary of g, as is guaranteed by Algorithm 1. Then, the safe boundary could be passed to a downstream optimiser that seeks to maximise f (either over all safe $(s, \mathbf{x})$, or over all s for each individual $\mathbf{x}$). This is akin to how the stage-wise algorithm in [Sui et al., 2018] operates.

Finally, more sophisticated algorithms may be possible that utilise information observed about f and g jointly throughout the course of the algorithm, but such investigations are left for future work.

Note on Contextual Settings: In some scenarios, it may be useful to incorporate *context variable* $\mathbf{c}$ in the problem setup, besides the *action variables* $s, \mathbf{x}$. For example, a clinical trial setting different from our previous motivating one might consist of choosing a drug dose s (with monotone behavior) and the dosage of a different drug $\mathbf{x}$ (not necessarily monotone), while also having access to patient characteristics $\mathbf{c}$ (e.g., BMI). All of these impact the drug toxicity, but in contrast to $(s, \mathbf{x})$, it may be that $\mathbf{c}$ cannot be chosen actively and is instead given by "nature".

In such scenarios, Algorithm 1 can be readily extended to incorporate context variables following the discussion on

contextual safe Bayesian optimisation from [Berkenkamp et al., 2021]. This approach allows joint modelling of the unknown function with the variables $(s, \mathbf{x}, \mathbf{c})$, so that information can be shared across contexts during optimisation. The proposed selection rule in this case would be: *for the given context* $\mathbf{c}_t$, search through the $\mathbf{x}$'s and find the highest possible safe s associated with each $\mathbf{x}$ (using the UCB as usual), and then among all candidate $(s, \mathbf{x})$'s, choose $(s_t, \mathbf{x}_t)$ to be the one with the highest posterior variance. Since the idea behind this extension closely follows [Berkenkamp et al., 2021], we do not elaborate on it further in this work.

4 THEORETICAL RESULTS

We present our main theoretical results in this section, under the set of assumptions outlined in 2. The proofs are provided in Appendix A. We note that in asymptotic statements, we treat the dimension d and sub-Gaussian parameter R (see (5)) as constants, and similarly treat the kernel as fixed.

Lemma 1. *(Theorem 2, [Chowdhury and Gopalan, 2017]) Fix $\delta > 0$, and suppose that β_t is set as follows:*

$$\beta_t = B + R\sqrt{2(\gamma_{t-1} + 1 + \ln(1/\delta))}. \tag{13}$$

Then, we have the following with probability at least $1 - \delta$:

$$|\mu_{t-1}(s, \mathbf{x}) - f(s, \mathbf{x})| \leq \beta_t \sigma_{t-1}(s, \mathbf{x}), \quad \forall \mathbf{x}, s, t, \tag{14}$$

where γ_t is the maximum information gain at time t:

$$\gamma_t := \max_{A \subset \mathcal{D}: |A| = t} I(y_A; f_A). \tag{15}$$

Here, $I(y_A; f_A)$ denotes the mutual information between $f_A = [f(\mathbf{x})]_{x \in A}$ and $y_A = f_A + \epsilon_A$, where $\epsilon_A \sim \mathcal{N}(0, \lambda I)$.

This lemma follows from Theorem 2 from [Chowdhury and Gopalan, 2017]. The quantity γ_t is ubiquitous in the GP bandit literature, and quantifies the maximum possible reduction in uncertainty about f after observing y_A at a set of points $A \subset \mathcal{D}$.

Theorem 1. *Under the setup and assumptions of Section 2, and the choice of β_t in Lemma 1,* M-SAFEUCB *satisfies the following regret bound with probability at least $1 - \delta$:*

$$R_T = O\left(B\sqrt{T\gamma_T} + \sqrt{T\gamma_T(\gamma_T + \ln(1/\delta))}\right). \tag{16}$$

As an example, with the squared exponential (SE) kernel on a compact subset $\mathcal{D} \subset \mathbb{R}^d$, γ_T is $O(\ln^{d+1} T)$ (Srinivas et al., 2010). Thus $R_T/T \to 0$ as $T \to \infty$, resulting in sublinear regret. The same holds for the Matérn kernel when the smoothness parameter ν is not too small.

The following theorem formalises the statement that the algorithm approximately identifies the entire safe region.

Theorem 2. *Consider the setup and assumptions of Section 2, and the choice of β_t in Lemma 1. With $l_h(s, \mathbf{x})$ as defined in equation* (3)*,* M-SAFEUCB *finds an ϵ-accurate solution $\hat{L}$ with probability at least $1 - \delta$:*

$$\mathbb{P}\left\{\max_{s, \mathbf{x} \in \mathcal{D}} l_h(s, \mathbf{x}) \leq \epsilon\right\} \geq 1 - \delta, \tag{17}$$

where ϵ scales as follows:

$$\epsilon = O\left(B\sqrt{\gamma_T/T} + \sqrt{(\gamma_T + \ln(1/\delta))\gamma_T/T}\right). \tag{18}$$

Substituting the bound on γ_T stated above for the squared exponential kernel (or the Matérn kernel when ν is not too small), we find that $\epsilon \to 0$ as $T \to \infty$.

The proof of Theorem 1 follows a similar general structure to regret analyses for GP-UCB and related algorithms [Srinivas et al., 2012, Chowdhury and Gopalan, 2017]. In contrast, Theorem 2 requires less standard techniques; see Appendix A for the complete argument.

Dependence on $\mathcal{D}_\mathcal{X}$: As we mentioned in Section 1, our theory circumvents strong dependencies on the domain size that are present in previous works for safe settings, and in fact holds even for continuous domains. The guarantees of SAFEOPT [Sui et al., 2015] (and related algorithms in follow-up works) roughly state that the entire reachable safe region, up to deviations of ϵ, will be identified with high probability once the time horizon T satisfies

$$\frac{T}{\beta_T \gamma_T} \geq \frac{C|\bar{R}_0|}{\epsilon^2}, \tag{19}$$

where C is a suitably-defined constant, and $\bar{R}_0 \subseteq \mathcal{D}_\mathcal{X}$ is a safe region that can potentially be reached from some initial safe seed set. While their setup is slightly different from ours (see Section 2), the associated guarantees readily transfer without significant modification. Under the mild assumption that $\bar{R}_0$ occupies a constant fraction of the domain, the requirement in (19) incurs a linear dependence on the domain size.

Our main results (Theorems 1 and 2) show that, in fact, the dependence on the domain size can be avoided altogether when we have the additional variable s that the function is monotone with respect to. We refer the reader to Appendix C.1 for further discussion in this regard.

Dependence on γ_T and T: Our regret bound in Theorem 1 incurs $\gamma_T\sqrt{T}$ dependence on T (up to logarithmic terms), and our convergence rate in Theorem 2 analogously incurs dependence $\frac{\gamma_T}{\sqrt{T}}$. This dependence matches that of GP-UCB [Srinivas et al., 2012] and other related algorithms for the standard (non-safe) setting, as well as SAFEOPT (and others) for the safe setting [Sui et al., 2015].

In the standard setting, it is known that the scaling can be improved to $\sqrt{\frac{\gamma_T}{T}}$ for simple regret [Vakili et al., 2021], and

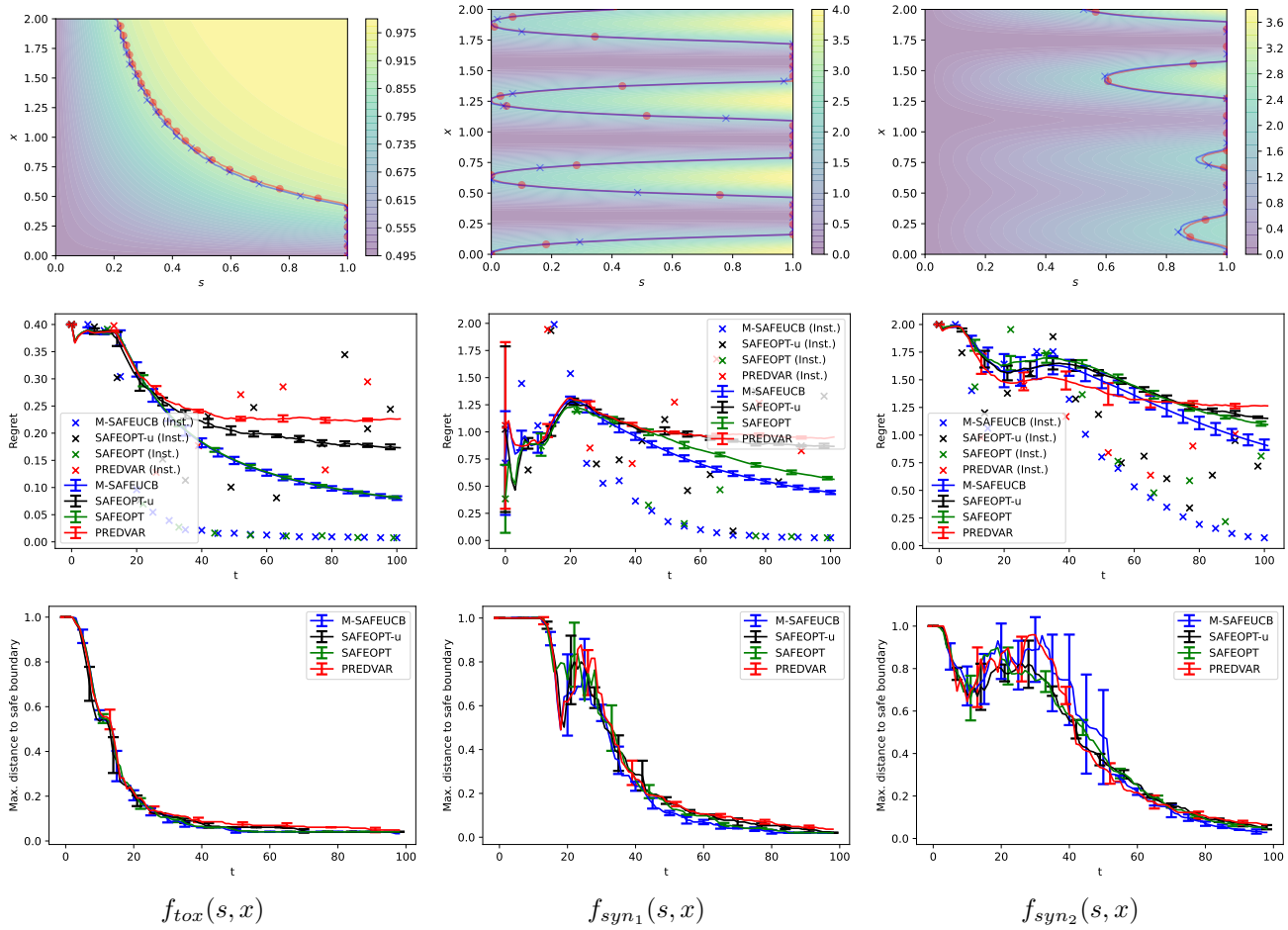

Figure 1: The results of running M-SAFEUCB on the functions f_{tox} (first column), f_{syn_1} (second column) and f_{syn_2} (third column). All functions are monotonically increasing in the input variable s. The first row shows the *safe boundary* as predicted by the algorithm (in blue) along with the actual safe boundary (in red), overlaid on the plot of the function. The second row shows the plots for the instantaneous and average cumulative regrets incurred by M-SAFEUCB, as well as SAFEOPT and PREDVAR (SAFEOPT-u denotes using an underestimate of the Lipschitz constant). The last row shows the maximum distance to the safe boundary across $x \in [0, 2]$ as a function of the time step. In each case, M-SAFEUCB is able to find the safe boundary almost exactly without sampling unsafe points, and the regret decreases towards zero.

$\sqrt{T\gamma_T}$ for cumulative regret [Li and Scarlett, 2022, Camilleri et al., 2021]; these improved bounds are near-optimal for common kernels such as Matérn. However, the techniques for attaining this improvement appear to be difficult to apply in the safe setting. For instance, the approaches of [Li and Scarlett, 2022] and [Camilleri et al., 2021] use a small number of batches (e.g., $O(\log T)$ or $O(\log\log T)$). In our setting, the safe set cannot be confidently expanded until the end of each batch, and this may be too infrequent to eventually find the safe boundary. In view of these difficulties, we believe that attaining near-optimal γ_T dependence in safe settings would be of significant interest for future work.

5 EXPERIMENTS

In this section, we present experimental results for M-SAFEUCB, and compare the performance to other representative algorithms for safe Bayesian optimisation [2]. The experiments serve to (i) investigate the cumulative regret of M-SAFEUCB and compare against baselines, (ii) compare the boundary of the sub-level set estimated by M-SAFEUCB with the actual boundary, and (iii) verify that unsafe points are not sampled during the optimisation. We emphasise that the main goal of this paper is not to have our algorithm outperform or "replace" any baselines. Rather, our main goal is to investigate the benefits of monotonicity, particularly from a theoretical standpoint. We provide the main information

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

regarding the functions, algorithms, and implementation here, and provide more details in Appendix B.

Simulated Clinical Trial: We first evaluate the performance of M-SAFEUCB on a synthetic function that simulates dose-toxicity response of certain drugs. We model toxicity using the logistic function following [Wang et al., 2022], and specifically consider the following:

$$f_{tox}(d, a) = \frac{1}{1 + e^{-\theta da}} \tag{20}$$

where d and a represent the drug dose (safety variable s) and the patient's age ($\mathbf{x}$) respectively. As discussed in Section 1, we consider the scenario where both the drug toxicity and efficacy increase monotonically with increasing dosage, and thus, the task reduces to finding the *maximum tolerated dose (MTD)*. We note that while the logistic function is often used to model dose-toxicity response, we incorporate the patient's age in the function here, such that the toxicity of a drug dose increases as the patient's age increases. This simulates the scenario that the *MTD* for a patient with a higher age is lower than that for a patient with a lower age. We set $\theta = 5$, the toxicity threshold $h = 0.9$, and the range of inputs is set as $d \in [0, 1], a \in [0, 2]$ (after suitable scale/shift).

We use the Matérn-$\frac{5}{2}$ kernel with trainable length-scale and variance parameters with log-normal priors. Based on minimal manual tuning and seeking simplicity, β_t is set to 5 and is kept constant throughout the optimisation (as in common in the Bayesian optimisation literature).

Fig. 1 shows the results obtained by running M-SAFEUCB, including the boundary of the safe set estimated, the regret incurred, and the worst-case (over $\mathbf{x}$) distance to the true safe boundary as a function of time. In each case, M-SAFEUCB succeeds in estimating the boundary very closely without sampling unsafe points, and the instantaneous regret goes to zero, indicating sub-linear cumulative regret.

We compare the performance of M-SAFEUCB with the SAFEOPT algorithm [Sui et al., 2015], and an active learning algorithm PREDVAR, that simply selects the point with the highest posterior variance (among those known to be safe) in each round [Schreiter et al., 2015]. We found that all algorithms maintain the safety requirement, and are roughly equally effective at identifying the safe region. In terms of regret, however, PREDVAR performs worse, and M-SAFEUCB tends to be best.

We note here that we ran the SAFEOPT by using the correct estimate of the Lipschitz constant L, as well as an underestimate, to demonstrate the effect on performance when a proper estimate of L is unavailable (SAFEOPT-u in Figure 1). We found that this can result in significantly degraded performance of SAFEOPT when L is underestimated (see Appendix B for more discussion on the effect of L). Further, SAFEOPT incurs a substantially higher computation time due to the requirement of finding the set of *potential expanders*.

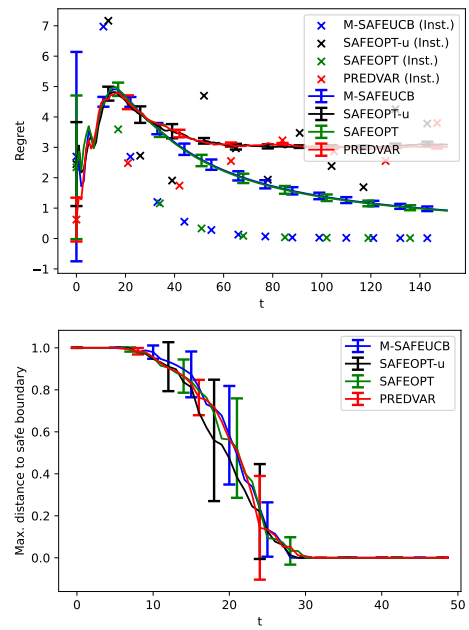

Figure 2: Results of running M-SAFEUCB, SAFEOPT and PREDVAR on the inverted pendulum swing-up problem. The top image shows the plots for the instantaneous and average cumulative regrets incurred by the algorithms. The bottom image shows the maximum distance to the safe boundary across x for the algorithms as a function of the time step.

Oscillating Synthetic Functions: Next, we evaluate the performance of M-SAFEUCB on two functions with a more complex form of the safe boundary. The functions are defined as follows:

$$\begin{aligned} f_{syn_1}(s, x) &= (1 + s)\,(1 + \cos(10x))\,, \\ f_{syn_2}(s, x) &= s\,(\exp(x)\sin(10x) + \sin(5x) + 5)\,/3. \end{aligned} \tag{21}$$

Both functions are monotonically increasing in the input variable s, and satisfy the remaining assumptions, i.e., $\forall x \in [0, 2], f(0, x) \leq h$ and $\max_{(s,x)\in\mathcal{D}} f(s, x) > h$, where the input range is set to $s \in [0, 1], x \in [0, 2]$ and the h is set to 2. We use the Matérn-$\frac{5}{2}$ kernel as earlier, and set β_t to 5 for f_{syn_1}, and 10 for f_{syn_2}.

Fig. 1 shows the results obtained by running M-SAFEUCB, as well as SAFEOPT and PREDVAR. Similar to the earlier results, M-SAFEUCB achieves sub-linear regret and finds the entire safe boundary while satisfying safety constraints. SAFEOPT also shows similar performance when a good estimate of L is available. PREDVAR and SAFEOPT-u perform worse in terms of regret, as can be expected.

Inverted Pendulum: We consider the *inverted pendulum swing-up* problem, a classic control task from the OpenAI

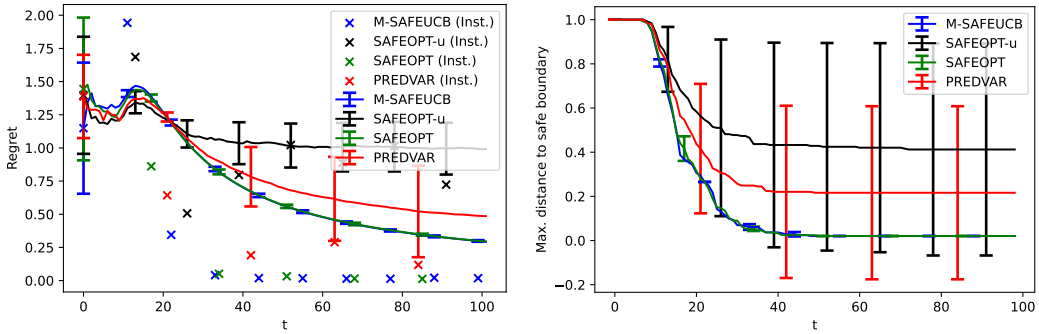

Figure 3: Results of running M-SafeUCB, SafeOpt and PredVar on the function $f_{syn_3}(s, \mathbf{x})$ given by (22). The left image shows the plots for the instantaneous and average cumulative regrets incurred by the three algorithms. The right image shows the maximum distance to the safe boundary across x for the three algorithms as a function of the time step.

Gym [Brockman et al., 2016] as a representative example of problems from robotics that fit into our problem setup. The goal of this task is to apply a torque to the free end of the pendulum to swing it to an upright position, starting from a random initial position. We modify the environment to suit the assumptions of our problem statement (see Appendix B for details). The algorithm is supposed to choose the initial torque that is applied to the pendulum, and the motion is simulated for 100 time steps. We modify the reward function as follows: (i) it equals the original reward if the pendulum does not cross the upright position, (ii) it equals zero if the pendulum reaches the upright position with zero angular velocity, and (iii) if the upright position is crossed, then we let the reward equal the angular velocity at the time of crossing. These changes are made primarily to ensure that the resulting reward function is smooth, and the action of not applying a torque ($s = 0$) is safe for all starting positions.

The goal is to maximise the reward function, while ensuring that the upright position is not crossed (thus, $h = 0$), i.e., case (ii) above is the optimal one, and case (iii) is unsafe.

The experimental results of running M-SafeUCB on this setup are presented in Figure 2, while comparing with SafeOpt and PredVar. We again observe strong similarities in terms of satisfying the safety threshold and finding the safe region. In this case we find that SafeOpt also closely matches M-SafeUCB in terms of regret, albeit at the cost of much higher computation time.

3D Input: Similar to the earlier experiments involving synthetic functions with 2D inputs, we evaluate the performance of M-SafeUCB on the following function with 3D inputs:

$$f_{syn_3}(s, \mathbf{x}) = s^2 + x_1^2 + x_2^2, \tag{22}$$

where s denotes the safety variable and $\mathbf{x} = (x_1, x_2)$ denotes the other input variable. The domain of each variable is set to $[0, 1]$, and the safety threshold is set to $h = 2$, thus satisfying the assumptions that for all $\mathbf{x} \in \mathcal{D}_{\mathcal{X}}$, $f(0, \mathbf{x}) \leq h$, and $\max_{(s,\mathbf{x})\in\mathcal{D}} f(s, \mathbf{x}) > h$. The average cumulative regret and the instantaneous regret for M-SafeUCB along with other baseline algorithms are shown in Figure 3. We mostly observe similar behavior to the 2D case, except that PredVar now incurs larger error bars.

Summary: Overall, our experimental results illustrate that (i) explicit expansion is not necessary under our assumed monotonicity conditions, and (ii) monotonicity not only benefits our proposed algorithm, but can also benefit other baselines and safe GP exploration methods in general.

6 CONCLUSION

We have demonstrated that monotonicity with respect to a single *safety variable* can have significant benefits for safe GP exploration and optimisation, including improved theoretical guarantees, algorithmic simplicity, and every safe point being reachable under mild conditions. Potential directions for future work include (i) seeking $\sqrt{\gamma_T}$ (rather than γ_T) dependence in the theoretical bounds, (ii) further studying more general scenarios with separate functions for safety and reward, (iii) determining other helpful function properties beyond monotonicity, and (iv) studying extensions to reinforcement learning settings, possibly either offline or online.

Acknowledgements

This work was supported by the Singapore National Research Foundation (NRF) under grant number A-0008064-00-00.

References

Sanae Amani, Mahnoosh Alizadeh, and Christos Thrampoulidis. Linear stochastic bandits under safety constraints. *Advances in Neural Information Processing Systems*, 32, 2019.

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.

Maryam Aziz, Emilie Kaufmann, and Marie-Karelle Riviere. On multi-armed bandit designs for dose-finding clinical trials. *Journal of Machine Learning Research*, 22:1–38, 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.

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.

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. Bayesian clinical trials. *Nature Reviews Drug Discovery*, 5(1):27–36, 2006.

Ilija Bogunovic, Jonathan Scarlett, Andreas Krause, and Volkan Cevher. Truncated variance reduction: A unified approach to Bayesian optimization and level-set estimation. *Advances in Neural Information Processing systems*, 29, 2016.

David Bolin and Finn Lindgren. Excursion and contour uncertainty regions for latent gaussian models. *Journal of the Royal Statistical Society: Series B: Statistical Methodology*, pages 85–106, 2015.

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

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

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

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

Christopher S Coffey and John A Kairalla. Adaptive clinical trials. *Drugs in R & D*, 9(4):229–242, 2008.

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

Kia Khezeli and Eilyan Bitar. Safe linear stochastic bandits. In *AAAI Conference on Artificial Intelligence*, volume 34, pages 10202–10209, 2020.

Hyun-Suk Lee, Cong Shen, James Jordon, and Mihaela Schaar. Contextual constrained learning for dose-finding clinical trials. In *International Conference on Artificial Intelligence and Statistics*, pages 2645–2654. PMLR, 2020.

Cheng Li, Santu Rana, Sunil Gupta, Vu Nguyen, and Svetha Venkatesh. Bayesian optimization with monotonicity information. In *NeurIPS Workshop on Bayesian Optimization*, 2017.

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

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.

Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2(117-129):2, 1978.

Marie-Karelle Riviere, Ying Yuan, Frédéric Dubois, and Sarah Zohar. A Bayesian dose-finding design for drug combination clinical trials based on the logistic model. *Pharmaceutical Statistics*, 13(4):247–257, 2014.

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.

Cong Shen, Zhiyang Wang, Sofia Villar, and Mihaela van der Schaar. Learning for dose allocation in adaptive clinical trials with safety constraints. In *International Conference on Machine Learning*, pages 8730–8740. PMLR, 2020.

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

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.

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.

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

Ami Takahashi and Taiji Suzuki. Bayesian optimization design for dose-finding based on toxicity and efficacy outcomes in phase I/II clinical trials. *Pharmaceutical Statistics*, 20(3):422–439, 2021a.

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

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294, 1933.

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

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.

Sattar Vakili, Nacime Bouziani, Sepehr Jalali, Alberto Bernacchia, and Da-shan Shiu. Optimal order simple regret for Gaussian process bandits. *Advances in Neural Information Processing Systems*, 34:21202–21215, 2021.

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.

W. Wang and W. J. Welch. Bayesian optimization using monotonicity information and its application in machine learning hyperparameter tuning. In *Proceedings of AutoML 2018 @ ICML/IJCAI-ECAI*, 2018.

Zhenlin Wang, Andrew J Wagenmaker, and Kevin Jamieson. Best arm identification with safety constraints. In *International Conference on Artificial Intelligence and Statistics*, pages 9114–9146. PMLR, 2022.