Safe Exploration for Interactive Machine Learning

Matteo Turchetta
Dept. of Computer Science
ETH Zurich
matteotu@inf.ethz.ch

Felix Berkenkamp
Dept. of Computer Science
ETH Zurich
befelix@inf.ethz.ch

Andreas Krause
Dept. of Computer Science
ETH Zurich
krausea@ethz.ch

Abstract

In Interactive Machine Learning (IML), we iteratively make decisions and obtain noisy observations of an unknown function. While IML methods, e.g., Bayesian optimization and active learning, have been successful in applications, on real-world systems they must provably avoid unsafe decisions. To this end, safe IML algorithms must carefully learn about *a priori* unknown constraints without making unsafe decisions. Existing algorithms for this problem learn about the safety of all decisions to ensure convergence. This is sample-inefficient, as it explores decisions that are not relevant for the original IML objective. In this paper, we introduce a novel framework that renders any existing unsafe IML algorithm safe. Our method works as an add-on that takes suggested decisions as input and exploits regularity assumptions in terms of a Gaussian process prior in order to efficiently learn about their safety. As a result, we only explore the safe set when necessary for the IML problem. We apply our framework to safe Bayesian optimization and to safe exploration in deterministic Markov Decision Processes (MDP), which have been analyzed separately before. Our method outperforms other algorithms empirically.

1 Introduction

Interactive Machine Learning (IML) problems, where an autonomous agent actively queries an unknown function to optimize it, learn it, or otherwise act based on the observations made, are pervasive in science and engineering. For example, Bayesian optimization (BO) (Mockus et al., 1978) is an established paradigm to optimize unknown functions and has been applied to diverse tasks such as optimizing robotic controllers (Marco et al., 2017) and hyperparameter tuning in machine learning (Snoek et al., 2012). Similarly, Markov Decision Processes (MDPs) (Puterman, 2014) model sequential decision making problems with long term consequences and are applied to a wide range of problems including finance and management of water resources (White, 1993).

However, real-world applications are subject to safety constraints, which cannot be violated during the learning process. Since the dependence of the safety constraints on the decisions is unknown *a priori*, existing algorithms are not applicable. To *optimize* the objective without violating the safety constraints, we must *carefully explore* the space and ensure that decisions are safe before evaluating them. In this paper, we propose a data-efficient algorithm for safety-constrained IML problems.

Related work One class of IML algorithms that consider safety are those for BO with Gaussian Process (GP) (Rasmussen, 2004) models of the objective. While classical BO algorithms focus on efficient optimization (Srinivas et al., 2010; Thompson, 1933; Wang and Jegelka, 2017), these methods have been extended to incorporate safety constraints. For example, Gelbart et al. (2014) present a variant of expected improvement with unknown constraints, while Hernández-Lobato et al. (2016) extend an information-theoretic BO criterion to handle black-box constraints. However, these methods only consider finding a safe solution, but allow unsafe evaluations during the optimization process. Wu et al. (2016) define safety as a constraint on the cumulative reward, while Schreiter et al. (2015) consider the safe exploration task on its own. The algorithms SAFEOPT (Sui et al., 2015;
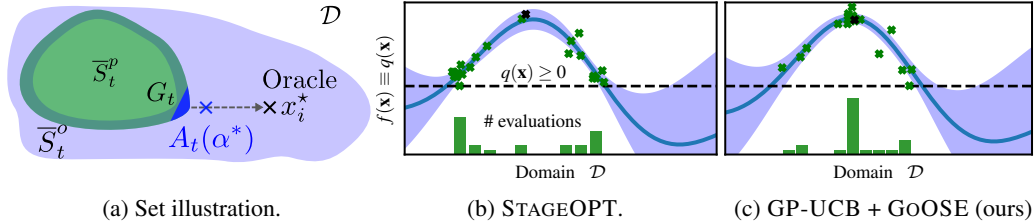
| (a) Set illustration. | (b) STAGEOPT. | (c) GP-UCB + GOOSE (ours) |

Figure 1: Existing algorithms for safe IML aim to expand the safe set $\bar{S}^p$ (green shaded) in Fig. 1a by evaluating decisions on the boundary of the pessimistic safe set (dark green shaded). This can be inefficient: to solve the safe BO problem in Fig. 1b, STAGEOPT evaluates decisions (green crosses, histogram) close to the safety constraint $q(\cdot) > 0$ (black dashed), even though the maximum (black cross) is known to be safe. In contrast, our method uses decisions $\mathbf{x}_i^\star$ from existing *unsafe* IML algorithms (oracle) within the optimistic safe set $\bar{S}_t^{o,\epsilon}$ (blue shaded, Fig. 1a). It can then use any heuristic to select learning targets A_t (blue cross) that are informative about the safety of $\mathbf{x}_i^\star$ and learns about them efficiently within $G_t \subseteq \bar{S}_t^p$ (blue shaded region). Since this method only learns about the safe set when necessary, we evaluate more close-to-optimal decisions in Fig. 1c.

Berkenkamp et al., 2016) and STAGEOPT (Sui et al., 2018) both guarantee safety of the exploration and near-optimality of the solution. However, they treat the exploration of the safe set as a proxy objective, which leads to sample-inefficient exploration as they explore the entire safe set, even if this is not necessary for the optimization task, see the evaluation counts (green) in Fig. 1b for an example.

Safety has also been investigated in IML problems in directed graphs, where decisions have long-term effects in terms of safety. Moldovan and Abbeel (2012) address this problem in the context of discrete MDPs by optimizing over ergodic policies, i.e., policies that are able to return to a known set of safe states with high probability. However, they do not provide exploration guarantees. Biyik et al. (2019) study the ergodic exploration problem in discrete and deterministic MDPs with unknown dynamics and noiseless observations. Turchetta et al. (2016) investigate the ergodic exploration problem subject to unknown external safety constraints under the assumption of known dynamics by imposing additional ergodicity constraints on the SAFEOPT algorithm. Wachi et al. (2018) compute approximately optimal policies in the same context but do not actively learn about the constraint. In continuous domains, safety has been investigated by, for example, Akametalu et al. (2014); Koller et al. (2018). While these methods provide safety guarantees, current exploration guarantees rely on uncertainty sampling on a discretized domain (Berkenkamp et al., 2017). Thus, their analysis can benefit from the more efficient, goal-oriented exploration introduced in this paper.

Contribution   In this paper, we introduce the Goal Oriented Safe Exploration algorithm, GOOSE; a novel framework that works as an add-on to existing IML algorithms and renders them safe. Given a possibly unsafe suggestion by an IML algorithm, it safely and efficiently learns about the safety of this decision by exploiting continuity properties of the constraints in terms of a GP prior. Thus, unlike previous work, GOOSE only learns about the safety of decisions relevant for the IML problem. We analyze our algorithm and prove that, with high probability, it only takes safe actions while learning about the safety of the suggested decisions. On safe BO problems, our algorithm leads to a bound on a natural notion of safe cumulative regret when combined with a no-regret BO algorithm. Similarly, we use our algorithm for the safe exploration in deterministic MDPs. Our experiments show that GOOSE is significantly more data-efficient than existing methods in both settings.

2   Problem Statement and Background

In IML, an agent iteratively makes decisions and observes their consequences, which it can use to make better decisions over time. Formally, at iteration i, the agent $\mathcal{O}_i$ uses the previous $i - 1$ observations to make a new decision $\mathbf{x}_i^\star = \mathcal{O}_i(\mathcal{D}_i)$ from a finite decision space $\mathcal{D}_i \subseteq \mathcal{D} \subseteq \mathbb{R}^d$. It then observes a noisy measurement of the unknown objective function $f : \mathcal{D} \to \mathbb{R}$ and uses the new information in the next iteration. This is illustrated in the top-left corner (blue shaded) of Fig. 2. Depending on the goal of the agent, this formulation captures a broad class of problems and many solutions to these problems have been proposed. For example, in *Bayesian optimization* the agent aims to find the global optimum $\max_{\mathbf{x}} f(\mathbf{x})$ (Mockus et al., 1978). Similarly, in active learning
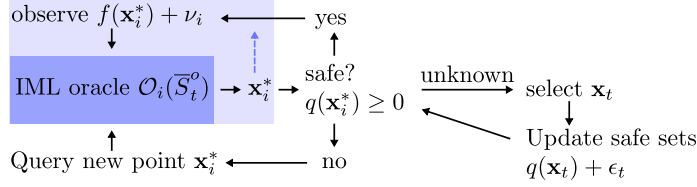
Figure 2: Overview of GoOSE. If the oracle's suggestion $\mathbf{x}_i^\star$ is safe, it can be evaluated. This is equivalent to the standard *unsafe* IML pipeline (top-left, blue shaded) in Fig. 2. Otherwise, GoOSE learns about the safety of $\mathbf{x}_i^\star$ by actively querying observations at decisions $\mathbf{x}_t$. Any provably unsafe decision is removed from the decision space and we query a new $\mathbf{x}_i^\star$ without providing a new observation of $f(\mathbf{x}_i^\star)$.

(Schreiter et al., 2015), one aims to learn about the function f. In the general case, the decision process may be stateful, e.g., as in dynamical systems, so that the decisions $\mathcal{D}_i$ available to the agent depend on those made in the past. This dependency among decisions can be modeled with a *directed graph*, where nodes represent decisions and an edge connects node $\mathbf{x}$ to node $\mathbf{x}'$ if the agent is allowed to evaluate $\mathbf{x}'$ given that it evaluated $\mathbf{x}$ at the previous decision step. In the BO setting, the graph is fully-connected and any decision may be evaluated, while in a deterministic MDP decisions are states and edges represent transitions (Turchetta et al., 2016).

In this paper, we consider IML problems with safety constraints, which frequently occur in real-world settings. The safety constraint can be written as $q(\mathbf{x}) \geq 0$ for some function q. Any decision $\mathbf{x}_i^\star$ for $i \geq 1$ evaluated by the agent must be safe. For example, Berkenkamp et al. (2016) optimize the control policy of a flying robot and must evaluate only policies that induce trajectories satisfying given constraints. However, it is unknown *a priori* which policy parameters induce safe trajectories. Thus, we do not know which decisions are safe in advance, that is, $q : \mathcal{D} \to \mathbb{R}$ is *a priori* unknown. However, we can learn about the safety constraint by selecting decisions $\mathbf{x}_t$ and obtaining noisy observations of $q(\mathbf{x}_t)$. We denote queries to f with $\mathbf{x}_i^\star$ and queries to q with $\mathbf{x}_t$. As a result, we face a two-tiered *safe exploration* problem: On one hand we have to safely learn about the constraint q to determine which decisions are safe, while on the other hand we want to learn about f to solve the IML problem. The goal is to minimize the number of queries $\mathbf{x}_t$ required to solve the IML problem.

Regularity Without further assumptions, it is impossible to evaluate decisions without violating the safety constraint q (Sui et al., 2015). For example, without an initial set of decisions that is known to be safe *a priori*, we may fail at the first step. Moreover, if the constraint does not exhibit any regularity, we cannot infer the safety of decisions without evaluating them first. We assume that a small initial safe set of decisions, S_0, is available, which may come from domain knowledge. Additionally, we assume that $\mathcal{D}$ is endowed with a positive definite kernel function, $k(\cdot, \cdot)$, and that the safety constraint q has bounded norm in the induced *Reproducing Kernel Hilbert Space* (RKHS) (Schölkopf and Smola, 2002)), $\|q\|_k \leq B_q$. The RKHS norm measures the smoothness of the safety feature with respect to the kernel, so q is L-Lipschitz continuous with respect to the kernel metric $d(\mathbf{x}, \mathbf{x}') = \sqrt{k(\mathbf{x}, \mathbf{x}) - 2k(\mathbf{x}, \mathbf{x}') + k(\mathbf{x}', \mathbf{x}')}$ with $L = B_q$ (Steinwart and Christmann, 2008, (4.21))

This assumption allows us to model the safety constraint function q with a GP (Rasmussen, 2004). A GP is a distribution over functions parameterized by a mean function $\mu(\cdot)$ and a covariance function $k(\cdot, \cdot)$. We set $\mu(\mathbf{x}) = 0$ for all $\mathbf{x} \in \mathcal{D}$ without loss of generality. The covariance function encodes our assumptions about the safety constraint. Given t observations of the constraint $\mathbf{y} = (q(\mathbf{x}_1) + \eta_1, \ldots, q(\mathbf{x}_t) + \eta_t)$ at decisions $\mathcal{D}_t = \{\mathbf{x}_n\}_{n=1}^t$, where $\eta_n \sim \mathcal{N}(0, \sigma^2)$ is a zero-mean i.i.d. Gaussian noise, the posterior belief is distributed as a GP with mean, covariance, and variance

$$\mu_t(\mathbf{x}) = \mathbf{k}_t^{\mathrm{T}}(\mathbf{x})(\mathbf{K}_t + \sigma^2\mathbf{I})^{-1}\mathbf{y}_t, \; k_t(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') - \mathbf{k}_t^{\mathrm{T}}(\mathbf{x})(\mathbf{K}_t + \sigma^2\mathbf{I})^{-1}\mathbf{k}_t(\mathbf{x}'), \; \sigma_t(\mathbf{x}) = k_t(\mathbf{x}, \mathbf{x})$$

respectively. Here, $\mathbf{k}_t(\mathbf{x}) = (k(\mathbf{x}_1, \mathbf{x}), \ldots, k(\mathbf{x}_t, \mathbf{x}))$, $\mathbf{K}_t$ is the positive definite kernel matrix $[k(\mathbf{x}, \mathbf{x}')]_{\mathbf{x}, \mathbf{x}' \in D_t}$, and $\mathbf{I} \in \mathbb{R}^{t \times t}$ denotes the identity matrix.

Safe decisions The previous regularity assumptions can be used to determine which decisions are safe to evaluate. Our classification of the decision space is related to the one by Turchetta et al. (2016), which combines non-increasing and reliable confidence intervals on q with a reachability analysis of the underlying graph structure for decisions. Based on a result by Chowdhury and Gopalan (2017), they use the posterior GP distribution to construct confidence bounds

3

$l_t(\mathbf{x}) := \max(l_{t-1}(\mathbf{x}), \mu_{t-1}(\mathbf{x}) - \beta_t \sigma_{t-1}(\mathbf{x}))$ and $u_t(\mathbf{x}) := \min(u_{t-1}(\mathbf{x}), \mu_{t-1}(\mathbf{x}) + \beta_t \sigma_{t-1}(\mathbf{x}))$ on the function q. In particular, we have $l_t(\mathbf{x}) \leq q(\mathbf{x}) \leq u_t(\mathbf{x})$ with high probability when the scaling factor β_t is chosen as in Theorem 1. Thus, any decision $\mathbf{x}$ with $l_t(\mathbf{x}) \geq 0$ is satisfies the safety constraint $q(\mathbf{x}) \geq 0$ with high probability.

To analyze the exploration behavior of their algorithm, Turchetta et al. (2016) use the confidence intervals within the current safe set, starting from S_0, and the Lipschitz continuity of q to define S_t^p, the set of decisions that satisfy the constraint with high probability. We use a similar, albeit more efficient, definition in Sec. 3. In practice, one may use the confidence intervals directly. Moreover, in order to avoid exploring decisions that are instantaneously safe but that would force the agent to eventually evaluate unsafe ones due to the graph structure $\mathcal{G}$, Turchetta et al. (2016) define $\bar{S}_t^p$, the subset of safe and ergodic decisions, i.e., decisions that are safe to evaluate in the short and long term.

Previous Exploration Schemes   Given that only decisions in $\bar{S}_t^p$ are safe to evaluate, any *safe* IML algorithm faces an extended exploration-exploitation problem: it can either optimize decisions within $\bar{S}_t^p$, or expand the set of safe decisions in $\bar{S}_t^p$ by evaluating decisions on its boundary. Existing solutions to the safe exploration problem in both discrete and continuous domains either do not provide theoretical exploration guarantees (Wachi et al., 2018) or treat the exploration of the safe set as a proxy objective for optimality. That is, the methods uniformly reduce uncertainty on the boundary of the safe set in Fig. 1a until the entire safe set is learned. Since learning about the entire safe set is often unnecessary for the IML algorithm, this procedure can be sample-inefficient. For example, in the safe BO problem in Fig. 1b with $f = q$, this exploration scheme leads to a large number of unnecessary evaluations on the boundary of the safe set.

3   Goal-oriented Safe Exploration (GOOSE)

In this section, we present our algorithm, GOOSE. We do not propose a new *safe* algorithm for a specific IML setting, but instead exploit that, for specific *IML* problems high-performance, *unsafe* algorithms already exist. We treat any such *unsafe* algorithm as an IML oracle $\mathcal{O}_i(S)$, which, given a domain S and $i - 1$ observations of f, suggests a new decision $\mathbf{x}_i^\star \in S$, see Fig. 2 (blue shaded).

GOOSE can extend any such *unsafe* IML algorithm to the safety-constrained setting. Thus, we effectively leave the problem of querying f to the oracle and only consider safety. Given an *unsafe* oracle decision $\mathbf{x}_i^\star$, GOOSE only evaluates $f(\mathbf{x}_i^\star)$ if the decisions $\mathbf{x}_i^\star$ is known to be safe. Otherwise, it *safely learns* about $q(\mathbf{x}_i^\star)$ by safely and efficiently collecting observations $q(\mathbf{x}_t)$. Eventually it either learns that the decision $\mathbf{x}_i^\star$ is safe and allows the oracle to evaluate $f(\mathbf{x}_i^\star)$, or that $\mathbf{x}_i^\star$ cannot be guaranteed to be safe given an ϵ-accurate knowledge of the constraint, in which case the decision set of the oracle is restricted and a new decision is queried, see Fig. 2.

Previous approaches treat the expansion of the safe set as a proxy-objective to provide completeness guarantees. Instead, GOOSE employs goal-directed exploration scheme with a novel theoretical analysis that shifts the focus from greedily reducing the uncertainty *inside* the safe set to learning about the safety of decisions *outside* of it. This scheme retains the worst-case guarantees of existing methods, but is significantly more sample-efficient in practice. Moreover, GOOSE encompasses existing methods for this problem. We now describe the detailed steps of GOOSE in Alg. 1 and 2.

Pessimistic and optimistic expansion.   To effectively shift the focus from inside the safe set to outside of it, GOOSE must reason not only about the decisions that are currently known to be safe but also about those that could eventually be classified as safe in the future. In particular, it maintains two sets, which are an inner/outer approximation of the set of safe decisions that are reachable from S_0 and are based on a pessimistic/optimistic estimate of the constraint given the data, respectively.

The pessimistic safe set contains the decisions that are safe with high probability and is necessary to guarantee safe exploration (Turchetta et al., 2016). It is defined in two steps: discarding the decisions that are not instantaneously safe and discarding those that we cannot reach/return from safely (see Fig. 3b) and, thus, are not safe in the long term. To characterize it starting from a given set of safe decisions S, we define the pessimistic constraint satisfaction operator,

$$p_t(S) = \{\mathbf{x} \in \mathcal{D}, \, | \, \exists \mathbf{z} \in S : l_t(\mathbf{z}) - Ld(\mathbf{x}, \mathbf{z}) \geq 0\}, \tag{1}$$

which uses the lower bound on the safety constraint of the decisions in S and the Lipschitz continuity of q to determine the decisions that instantaneously satisfy the constraint with high probability, see Fig. 3a. However, for a general graph $\mathcal{G}$, decisions in $p_t(S)$ may be unsafe in the long-term

4

(a) Expansion operators.
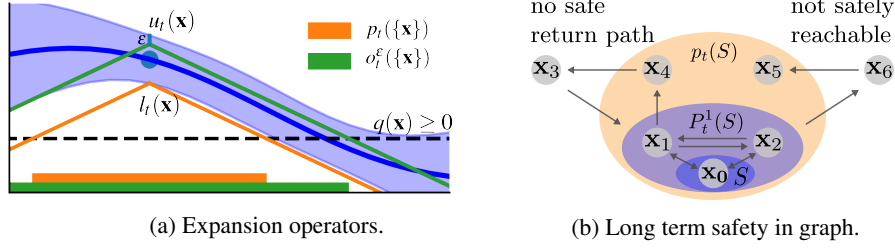

(b) Long term safety in graph.

Figure 3: Fig. 3a shows the pessimistic and optimistic constraint satisfaction operators that use the confidence intervals on the constraint and its Lipschitz continuity to make inference about the safety of decisions that have not yet been evaluated. Fig. 3b illustrates the long-term safety definition. While decisions in $p_t(S)$ are myopically safe, decisions in $P_t^1(S)$ are safe in the long-term. This excludes $\mathbf{x}_4$ and $\mathbf{x}_5$, as no safe path from/to them exists.

as in Fig. 3b: No safe path to the decision $\mathbf{x}_5$ exists, so that it can not be safely *reached*. Similarly, if we were to evaluate $\mathbf{x}_4$, the graph structure forces us to eventually evaluate $\mathbf{x}_3$, which is not contained in $p_t(S)$ and might be unsafe. That is, we cannot safely *return* from $\mathbf{x}_4$. To exclude these decisions, we use the ergodicity operator introduced by Turchetta et al. (2016), which allows us to find those decisions that are pessimistically safe in the short and long term $P_t^1(S) = p_t(S) \cap R^{\mathrm{ergodic}}(p_t(S), S)$ (see Appendix A or (Turchetta et al., 2016) for the definition of R^{ergodic}). Alternating these operations n times, we obtain the n-step pessimistic expansion operator, $P_t^n(S) = p_t(P_t^{n-1}(S)) \cap R^{\mathrm{ergodic}}(p_t(P_t^{n-1}(S)), S)$, which, after a finite number of steps, converges to its limiting set $\tilde{P}_t(S) = \lim_{n\to\infty} P_t^n(S)$.

The optimistic safe set excludes the decisions that are unsafe with high probability and makes the exploration efficient by restricting the decision space of the oracle. Similarly to the pessimistic one, it is defined in two steps. However, it uses the following optimistic constraint satisfaction operator,

$$o_t^\epsilon(S) = \{\mathbf{x} \in \mathcal{D}, \,|\, \exists \mathbf{z} \in S : u_t(\mathbf{z}) - Ld(\mathbf{x}, \mathbf{z}) - \epsilon \geq 0\}. \tag{2}$$

See Fig. 3a for a graphical intuition. The additional ϵ-uncertainty term in the optimistic operator accounts for the fact that we only have access to noisy measurements of the constraint and, therefore, we can only learn it up to a specified statistical accuracy. The definitions of the optimistic expansion operators $O_t^{\epsilon,n}(S)$ and $\tilde{O}_t^\epsilon(S)$ are analogous to the pessimistic case by substituting p_t with o_t^ϵ. The sets $\tilde{P}_t$ and $\tilde{O}_t^\epsilon$ indicate the largest set of decisions that can be classified as safe in the short and long term assuming the constraint attains the worst/best possible value within S, given the observations available and, for the optimistic case, despite an ϵ uncertainty.

Optimistic oracle The IML oracle $\mathcal{O}_i(S)$ suggests decisions $\mathbf{x}_i^\star \in S$ to evaluate within a given subset S of $\mathcal{D}$. To make the oracle efficient, we restrict its decision space to decisions that could optimistically be safe in the long and short term. In particular, we define the optimistic safe set $\bar{S}^{o,\epsilon}$ in Line 8 of Alg. 1 based on the optimistic expansion operator introduced above. The oracle uses this set to suggest a potentially unsafe, candidate decision $\mathbf{x}_i^\star = \mathcal{O}_i(\bar{S}_t^{o,\epsilon})$ in Line 4.

Safe evaluation We determine safety of the suggestion $\mathbf{x}_i^\star$ similarly to Turchetta et al. (2016) by constructing the set $\bar{S}_t^p$ of decisions that are safe to evaluate. However, while Turchetta et al. (2016) use the one step pessimistic expansion operator in their definition, P_t^1, we use its limit set in Line 7 of Alg. 1, $\tilde{P}_t$. While both operators eventually identify the same safe set, our definition allows for a more efficient expansion. For example, consider the case where the graph over the decision space $\mathcal{G}$ is a chain of length m and where, for all $j = 1, \cdots, m$, the lower bound on the safety of decision $j - 1$ guarantees the safety of decision j with high probability. In this case, Turchetta et al. (2016) require $m - 1$ iterations to fully expand the safe set, while our classification requires only one.

If we know that $\mathbf{x}_i^\star$ is safe to evaluate, i.e., $\mathbf{x}_i^\star \in \bar{S}_t^p$, then the oracle obtains a noisy observation of $f(\mathbf{x}_i^\star)$ in Line 10. Otherwise GOOSE proceeds to safely learn about the safety of $\mathbf{x}_i^\star$ using a safe expansion strategy in lines Lines 5–8 that we outline in the following. This routine is repeated until we can either include $\mathbf{x}_i^\star$ in $\bar{S}_t^p$, in which case we can safely evaluate $f(\mathbf{x}_i^\star)$, or remove it from the decision space $\bar{S}_t^{o,\epsilon}$ and query the oracle for a new suggestion.

5

Algorithm 1 GoOSE

1: **Inputs:** Lipschitz constant L, Seed S_0,
   Graph $\mathcal{G}$, Oracle O, Accuracy ϵ.
2: $\bar{S}_0^p \leftarrow S_0, \bar{S}_0^{o,\epsilon} \leftarrow \mathcal{D}, t \leftarrow 0,$
   $l_0(\mathbf{x}) \leftarrow 0$ for $\mathbf{x} \in S_0$
3: **for** $k = 1, 2, \ldots$ **do**
4: $\quad \mathbf{x}_i^\star \leftarrow \mathcal{O}(\bar{S}_t^{o,\epsilon})$
5: $\quad$ **while** $\mathbf{x}_i^\star \notin \bar{S}_t^p$ **do**
6: $\quad\quad$ SE$(\bar{S}_t^{o,\epsilon}, \bar{S}_t^p, \mathcal{G}, \mathbf{x}_i^\star), t \leftarrow t+1$
7: $\quad\quad \bar{S}_t^p \leftarrow \tilde{P}_t(\bar{S}_{t-1}^p)$
8: $\quad\quad \bar{S}_t^{o,\epsilon} \leftarrow \tilde{O}_t^\epsilon(\bar{S}_{t-1}^p)$
9: $\quad\quad$ **if** $\mathbf{x}_i^\star \notin \bar{S}_t^{o,\epsilon}$ **then go to** Line 4
10: $\quad$ Evaluate $f(\mathbf{x}_i^\star)$ and update oracle

Algorithm 2 Safe Expansion (SE)

1: **Inputs:** $\bar{S}_t^{o,\epsilon}, \bar{S}_t^p, \mathcal{G}, \mathbf{x}^\star$
2: $W_t^\epsilon \leftarrow \{\mathbf{x} \in \bar{S}_t^p \,|\, u_t(\mathbf{x}) - l_t(\mathbf{x}) > \epsilon\}$
3: $A_t(p) \leftarrow \{\mathbf{x} \in \bar{S}_t^{o,\epsilon} \setminus p_t^0(\bar{S}_t^p) \,|\, h(\mathbf{x}) = p\}$
4: // Highest priority targets in A_t with expanders
   $\alpha^* \leftarrow \max \alpha$ s.t. $|G_t^\epsilon(\alpha)| > 0$
5: **if** optimization problem feasible **then**
6: $\quad \mathbf{x}_t \leftarrow \operatorname{argmax}_{\mathbf{x} \in G_t^\epsilon(\alpha^*)} w_t(\mathbf{x})$
7: $\quad$ Update GP with $y_t = q(\mathbf{x}_t) + \eta_t$

Safe expansion If the oracle suggestion $\mathbf{x}_i^\star$ is not considered safe, $\mathbf{x}_i^\star \notin \bar{S}_t^p$, GoOSE employs a goal-directed scheme to evaluate a safe decision $\mathbf{x}_t \in \bar{S}_t^p$ that is informative about $q(\mathbf{x}_i^\star)$, see Fig. 1a. In practice, it is desirable to avoid learning about decisions beyond a certain accuracy ϵ, as the number of observations required to reduce the uncertainty grows exponentially with ϵ (Sui et al., 2018). Thus, we only learn about decisions in $\bar{S}_t^p$ whose safety values are not known ϵ-accurately yet in Line 2, $W_t^\epsilon = \{\mathbf{x} \in \bar{S}_t^p \,|\, u_t(\mathbf{x}) - l_t(\mathbf{x}) > \epsilon\}$, where $u_t(\mathbf{x}) - l_t(\mathbf{x})$ is the width of the confidence interval at $\mathbf{x}$.

To decide which decision in W_t^ϵ to learn about, we first determine a set of learning targets outside the safe set (dark blue cross in Fig. 1a), and then learn about them efficiently within $\bar{S}_t^p$. To quantify how useful a learning target $\mathbf{x}$ is to learn about $q(\mathbf{x}_i^\star)$, we use any given iteration-dependent heuristic $h_t(\mathbf{x})$. We discuss particular choices later, but a large priority $h(\mathbf{x})$ indicates a relevant learning target (dashed line, Fig. 1a). Since $p_t^0(\bar{S}_t^p)$ denotes the decisions that are known to satisfy the constraint with high probability and $\bar{S}_t^{o,\epsilon}$ excludes the decisions that are unsafe with high probability, $\bar{S}_t^{o,\epsilon} \setminus p_t^0(\bar{S}_t^p)$ indicates the decisions whose safety we are uncertain about. We sort them according to their priority and let $A_t(\alpha)$ denote the subset of decision with equal priority.

Ideally, we want to learn about the decisions with the highest priority. However, this may not be immediately possible by evaluating decisions within W_t^ϵ. Thus, we must identify the decisions with the highest priority that we can learn about starting from W_t^ϵ. Therefore, similarly to the definition of the optimistic safe set, we identify decisions $\mathbf{x}$ in W_t^ϵ that have a large enough plausible value $q(\mathbf{x})$ that they could guarantee that $q(\mathbf{z}) \geq 0$ for some $\mathbf{z}$ in $A_t(\alpha)$. However, in this case, we are only interested in decisions that can be instantly classified as safe (rather than eventually). Therefore, we focus on this set of *potential immediate expanders*, $G_t^\epsilon(\alpha) = \{\mathbf{x} \in W_t^\epsilon, \,|\, \exists \mathbf{z} \in A_t(\alpha) \colon u_t(\mathbf{x}) - Ld(\mathbf{x}, \mathbf{z}) \geq 0\}$. In Line 4 of Alg. 2 we select the decisions with the priority level α^* such that there exist uncertain, safe decisions in W_t^ϵ that could allow us to classify a decision in $A_t(\alpha^*)$ as safe and thereby expand the current safe set $\bar{S}_t^p$. Intuitively, we look for the highest priority targets that can potentially be classified as safe by safely evaluating decisions that we have not already learned about to ϵ-accuracy.

Given these learning targets $A_t(\alpha^*)$ (blue cross, Fig. 1a), we evaluate the most uncertain decision in $G_t^\epsilon(\alpha^*)$ (blue shaded, Fig. 1a) in Line 6 and update the GP model with the corresponding observation of $q(\mathbf{x}_t)$ in Line 7. This uncertainty sampling is restricted to a small set of decisions close to the goal. This is different from methods without a heuristic that select the most uncertain secision on the boundary of $\bar{S}^p$ (green shaded in Fig. 1a). In fact, our method is equivalent to the one by Turchetta et al. (2016) when an uninformative heuristic $h(\mathbf{x}) = 1$ is used for all $\mathbf{x}$. We iteratively select and evaluate decisions $\mathbf{x}_t$ until we either determine that $\mathbf{x}_i^\star$ is safe, in which case it is added to $\bar{S}^p$, or we prove that we can not safely learn about it for given accuracy ϵ, in which case is removed from $\bar{S}^{o,\epsilon}$ and a the oracle is queried with an updated decision space for a new suggestion.

To analyze our algorithm, we define the largest set that we can learn about as $\tilde{R}_\epsilon(S_0)$. This set contains all the decisions that we could certify as safe if we used a full-exploration scheme that learns the safety constraint q up to ϵ accuracy for all decisions inside the current safe set. This is a natural exploration target for our safe exploration problem (see Appendix A for a formal definition). We have the following main result, which holds for any heuristic:

Theorem 1. *Assume that $q(\cdot)$ is L-Lipschitz continuous w.r.t. $d(\cdot, \cdot)$ with $\|q\|_k \le B_q$, σ-sub-Gaussian noise, $S_0 \ne \emptyset$, $q(\boldsymbol{x}) \ge 0$ for all $\boldsymbol{x} \in S_0$, and that, for any two decisions $\boldsymbol{x}, \boldsymbol{x}' \in S_0$, there is a path in the graph $\mathcal{G}$ connecting them within S_0. Let $\beta_t^{1/2} = B_q + 4\sigma\sqrt{\gamma_t + 1 + \ln(1/\delta)}$, then, for any $h_t : \mathcal{D} \to \mathbb{R}$, with probability at least $1 - \delta$, we have $q(\boldsymbol{x}) \ge 0$ for any $\boldsymbol{x}$ visited by GOOSE. Moreover, let γ_t denote the information capacity associated with the kernel k and let t^* be the smallest integer such that $\frac{t^*}{\beta_{t^*}\gamma_{t^*}} \ge \frac{C|\tilde{R}_0(S_0)|}{\epsilon^2}$, with $C = 8/\log(1 + \sigma^{-2})$, then there exists a $t \le t^*$ such that, with probability at least $1 - \delta$, $\tilde{R}_\epsilon(S_0) \subseteq \bar{S}_t^{o,\epsilon} \subseteq \bar{S}_t^p \subseteq \tilde{R}_0(S_0)$.*

Theorem 1 guarantees that GOOSE is safe with high probability. Moreover, for any priority function h in Alg. 2, it upper bounds the number of measurements that Alg. 1 requires to explore the largest safely reachable region $\tilde{R}_\epsilon(S_0)$. Note that GOOSE only achieves this upper bound if it is required by the IML oracle. In particular, the following is a direct consequence of Theorem 1:

Corollary 1. *Under the assumptions of Theorem 1, let the IML oracle be deterministic given the observations. Then there exists a set S with $\tilde{R}_\epsilon(S_0) \subseteq S \subseteq \tilde{R}_0(S_0)$ so that $\boldsymbol{x}_i^\star = \mathcal{O}(S)$ for all $k \ge 1$.*

That is, the oracle decisions $\mathbf{x}_i^\star$ that we end up evaluating are the same as those by an oracle that was given the safe set S in Corollary 1 from the beginning. This is true since the set $\bar{S}_t^{o,\epsilon}$ converges to this set S. Since Theorem 1 bounds the number of safety evaluations by t^*, Corollary 1 implies that, up to t^* safety evaluations, GOOSE retains the properties (e.g., no-regret) of the IML oracle $\mathcal{O}$ over S.

Choice of heuristic While our worst-case guarantees hold for *any* heuristic, the empirical performance of GOOSE depends on this choice. We propose to use the graph structure directly and additionally define a positive cost for each edge between two nodes. For a given edge cost, we define $c(\mathbf{x}, \mathbf{x}_k^\star, \bar{S}_t^{o,\epsilon})$ as the cost of the minimum-cost path from $\mathbf{x}$ to $\mathbf{x}_k^\star$ within the optimistic safe set $\bar{S}_t^{o,\epsilon}$, which is equal to ∞ if a path does not exist, and we consider the priority $h(\mathbf{x}) = -c(\mathbf{x}, \mathbf{x}_k^\star, \bar{S}_t^{o,\epsilon})$. Thus, the node $\mathbf{x}$ with the lowest-cost path to $\mathbf{x}_k^\star$ has the highest priority. This reduces the design of a general heuristic to a more intuitive weight assignment problem, where the edge costs determine the planned path for learning about $\mathbf{x}_k^\star$ (dashed line in Fig. 1a). One option for the edge cost is the inverse mutual information between $\mathbf{x}$ and the suggestion $\mathbf{x}_k^\star$, so that the resulting paths contain nodes that are informative about $\mathbf{x}_k^\star$. Alternatively, having successive nodes in the path close to each other under the metric $d(\cdot, \cdot)$, so that they can be easily added to the safe set and eventually lead us to $\mathbf{x}_k^\star$, can be desirable. Thus, increasing monotone functions of the metric $d(\cdot, \cdot)$ can be effective edge costs.

4 Applications and Experiments

In this section, we introduce two safety-critical IML applications, discuss the consequences of Theorem 1 for these problems, and empirically compare GOOSE to stae-of-the-art competing methods. In our experiments, we set $\beta_t = 3$ for all $t \ge 1$ as suggested by Turchetta et al. (2016). This choice of β_t ensures safety in practice, but leads to more efficient exploration than the theoretical choice in Theorem 1 (Turchetta et al., 2016; Wachi et al., 2018). Moreover, since in practice it is hard to estimate the Lipschitz constant of an unknown function, in our experiments we use the confidence intervals to define the safe set and the expanders as suggested by Berkenkamp et al. (2016).

4.1 Safe Bayesian optimization

In safe BO we want to optimize the unknown function f subject to the unknown safety constraint q, see Sec. 2. In this setting, we aim to find the best input over the largest set we can hope to explore safely, $\tilde{R}_\epsilon(S_0)$. The performance of an agent is measured in terms of the ϵ-safe regret $\mathrm{argmax}_{\mathbf{x} \in \tilde{R}_\epsilon(S_0)} f(\mathbf{x}) - f(\mathbf{x}_t)$ of not having evaluated the function at the optimum in $\tilde{R}_\epsilon(S_0)$.

We combine GOOSE with the unsafe GP-UCB (Srinivas et al., 2010) algorithm as an oracle. For computational efficiency, we do not use a fully connected graph, but instead connect decisions only to their immediate neighbors as measured by the kernel and assign equal weight to each edge for the heuristic h. We compare GOOSE to SAFEOPT (Sui et al., 2015) and STAGEOPT (Sui et al., 2018) in terms of ϵ-*safe average regret*. Both algorithms use safe exploration as a proxy objective, see Fig. 1.

We optimize samples from a GP with zero mean and Radial Basis Function (RBF) kernel with variance 1.0 and lengthscale 0.1 and 0.4 for a one-dimensional and two-dimensional, respectively.
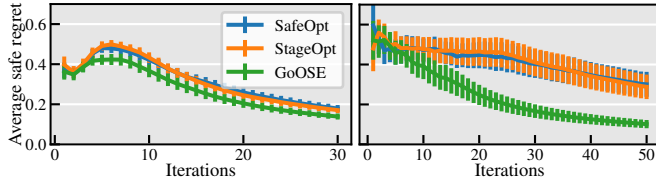
Figure 4: Average normalized ϵ-safe regret for the safe optimization of GP samples over 40 (d=1, left) and 10 (d=2, right) samples. GOOSE only evaluates inputs that are relevant for the BO problem and, therefore, it converges faster than its competitors.

Table 1: Mars experiment performance normalized to SMDP in terms of samples to find the first path, exploration cost and computation time per iteration.

	GOOSE	SEO
Sample	**30.0** %	38.4 %
Cost	12.7 %	**0.7** %
Time	**37.8** %	518 %

(a) Cost of exploration.

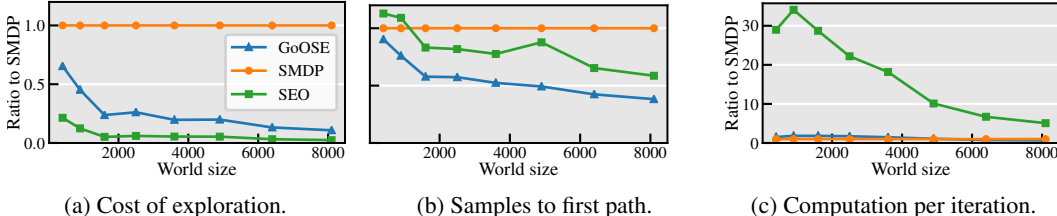(b) Samples to first path.

(c) Computation per iteration.

Figure 5: Performance of GOOSE and SEO normalized to SMDP in terms of exploration cost, samples to find the first path and computation time per iteration as a function of the world size.

The observations are perturbed by i.i.d Gaussian noise with $\sigma = 0.01$. For simplicity, we set the objective and the constraint to be the same, $f = q$. Fig. 4 (left) shows the average regret as a function of the number of evaluations $k + t$ averaged over 40 different samples from the GP described above over a one dimensional domain (200 points evenly distributed in $[-1, 1]$). Fig. 4 (right) shows similar results averaged over 10 samples for a two dimensional domain (25×25 uniform grid in $[0, 1]^2$).

These results confirm the intuition from Fig. 1 that using safe exploration as a proxy objective reduces the empirical performance of safe BO algorithms. The impact is more evident in the two dimensional case where there are more points along the boundaries that are nor relevant to the optimization and that are evaluated for exploration purposes.

4.2 Safe shortest path in deterministic MDPs

The graph that we introduced in Sec. 2 can model states (nodes) and state transitions (edges) in deterministic, discrete MDPs. Hence, GOOSE naturally extends to the goal-oriented safe exploration problem in these models. We aim to find the minimum-cost safe path from a starting state $\mathbf{x}^\dagger$ to a goal state $\mathbf{x}^\star$, without violating the unknown safety constraint, q. At best, we can hope to find the path within the largest safely learnable set $\tilde{R}_\epsilon(S_0)$ as in Theorem 1 with cost $c(\mathbf{x}^\dagger, \mathbf{x}^\star, \tilde{R}_\epsilon(S_0))$.

Algorithms   We compare GOOSE to SEO (Wachi et al., 2018) and SMPD (Turchetta et al., 2016) in terms of samples required to discover the first path, total exploration cost and computation cost on synthetic and real world data. The SMDP algorithm cannot take goals into account and serves as a benchmark for comparison. The SEO algorithm aims to safely learn a near-optimal policy for any given cost function and can be adapted to the safe shortest path problem by setting the cost to $c(\mathbf{x}) = -\|\mathbf{x} - \mathbf{x}^\star\|_1$. However, it cannot guarantee that a path to $\mathbf{x}^\star$ is found, if one exists. Since the goal $\mathbf{x}^\star$ is fixed, GOOSE does not need an oracle. For the heuristic we use and optimistic estimate of the cost of the safe shortest path from $\mathbf{x}^\dagger$ to $\mathbf{x}^\star$ passing through $\mathbf{x}$; that is $h_t(\mathbf{x}) = -\min_{\mathbf{x}' \in Pred(\mathbf{x})} c(\mathbf{x}^\dagger, \mathbf{x}, \bar{S}_t^p) + \kappa c(\mathbf{x}, \mathbf{x}^\star, \bar{S}_t^{o,\epsilon})$. The first term is a conservative estimate of the safe optimal cost from $\mathbf{x}^\dagger$ to the best predecessor of $\mathbf{x}$ in $\mathcal{G}$ and the second term is an optimistic estimate of the safe optimal cost from $\mathbf{x}$ to $\mathbf{x}^\star$ multiplied by $\kappa > 1$ to encourage early path discovery. Here, we use the predecessor node because $\phi_t(\mathbf{x}) = \infty$ for all $\mathbf{x}$ not in $\bar{S}_t^p$. Notice that, if a safe path exists, Theorem 1 guarantees that GOOSE finds the shortest one eventually.

Synthetic data   Similarly to the setting in Turchetta et al. (2016); Wachi et al. (2018) we construct a two-dimensional grid world. At every location, the agent takes one of four actions: *left*, *right*, *up* and *down*. We use the state augmentation in Turchetta et al. (2016) to define a constraint over state

transitions. The constraint function is a sample from a GP with mean $\mu = 0.6$ and RBF kernel with lengthscale $l = 2$ and variance $\sigma^2 = 1$. If the agent takes an unsafe action, it ends up in a failure state, otherwise it moves to the desired adjacent state. We make the constraint independent of the direction of motion, i.e., $q(\mathbf{x}, \mathbf{x}') = q(\mathbf{x}', \mathbf{x})$. We generate 800 worlds by sampling 100 different constraints for square maps with sides of $20, 30, 40, \cdots, 90$ tiles and a source-target pair for each one.

We show the geometric mean of the performance of SEO and GOOSE relative to SMDP as a function of the world size in Fig. 5. Fig. 5b shows that GOOSE needs a factor 2.5 fewer samples than SMDP. Fig. 5c shows that the overhead to compute the heuristic of GOOSE is negligible, while the solution of the two MDPs [1] required by SEO is computationally intense. Fig. 5a shows that SEO outperforms GOOSE in terms of cost of the exploration trajectory. This is expected as SEO aims to minimize it, while GOOSE optimizes the sample-efficiency. However, it is easy to modify the heuristic of GOOSE to consider the exploration cost by, for example, reducing the priority of a state based on its distance from the current location of the agent. In conclusion, GOOSE leads to a drastic improvement in performance with respect to the previously known safe exploration strategy with exploration guarantees, SMDP. Moreover, it achieves similar or better performance than SEO while providing exploration guarantees that SEO lacks.

Mars exploration We simulate the exploration of Mars with a rover. In this context, communication delays between the rover and the operator on Earth make autonomous exploration extremely important, while the high degree of uncertainty about the environment requires the agent to consider safety constraints. In our experiment, we consider the *Mars Science Laboratory* MSL (2007, Sec. 2.1.3), a rover deployed on Mars that can climb a maximum slope of $30°$. We use Digital Terrain Models of Mars available from McEwen et al. (2007).

We use a grid world similar to the one introduced above. The safety constraint is the absolute value of the steepness of the slope between two locations: given two states $\mathbf{x}$ and $\mathbf{x}'$, the constraint over the state transition is defined as $q(\mathbf{x}, \mathbf{x}') = |H(\mathbf{x}) - H(\mathbf{x}')|/d(\mathbf{x}, \mathbf{x}')$, where $H(\mathbf{x}), H(\mathbf{x}')$ indicate the altitudes at $\mathbf{x}$ and $\mathbf{x}'$ respectively and $d(\mathbf{x}, \mathbf{x}')$ is the distance between them. We set conservatively the safety constraint to $q(\mathbf{x}, \mathbf{x}') \geq -\tan^{-1}(25°)$. The step of the grid is 10m. We use square maps from 16 different locations on Mars with sides between 100 and 150 tiles. We generate 64 scenarios by sampling 4 source-target pairs for each map . We model the steepness with a GP with Matérn kernel with $\nu = 5/2$. We set the hyperprior on the lengthscale and on the standard deviation to be $Lognormal(30\text{m}, 0.25\text{m}^2)$ and $Lognormal(\tan(10°), 0.04)$, respectively. These encode our prior belief about the surface of Mars. Next, we take 1000 noisy measurements at random locations from each map, which, in reality, could come from satellite images, to find a maximum a posteriori estimator of the hyperparameters to fine tune our prior to each site.

In Tab. 1, we show the geometric mean of the performance of SEO and GOOSE relative to SMDP. The results confirm those of the synthetic experiments but with larger changes in performance with respect to the benchmark due to the increased size of the world.

5   Conclusion

We presented GOOSE, an add-on module that enables existing interactive machine learning algorithms to safely explore the decision space, without violating *a priori* unknown safety constraints. Our method is provably safe and learns about the safety of decisions suggested by existing, unsafe algorithms. As a result, it is more data-efficient than previous safe exploration methods in practice.

---

[1] We use policy iteration. Policy evaluation is performed by solving a sparse linear system with SciPy (Virtanen et al., 2019). At iteration t, we initialize policy iteration with the optimal policy from $t - 1$.

References

Anayo K Akametalu, Jaime F Fisac, Jeremy H Gillula, Shahab Kaynama, Melanie N Zeilinger, and Claire J Tomlin. Reachability-based safe learning with gaussian processes. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pages 1424–1431. IEEE, 2014.

Felix Berkenkamp, Andreas Krause, and Angela P Schoellig. Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics. *arXiv preprint arXiv:1602.04450*, 2016.

Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. In *Advances in Neural Information Processing Systems*, pages 908–919, 2017.

Erdem Biyik, Jonathan Margoliash, Shahrouz R. Alimo, and Dorsa Sadigh. Efficient and safe exploration in deterministic markov decision processes with unknown transition models. In *Proceedings of the American Control Conference (ACC)*, July 2019.

Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 844–853, 2017.

Michael A Gelbart, Jasper Snoek, and Ryan P Adams. Bayesian optimization with unknown constraints. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pages 250–259. AUAI Press, 2014.

José Miguel Hernández-Lobato, Michael A Gelbart, Ryan P Adams, Matthew W Hoffman, and Zoubin Ghahramani. A general framework for constrained bayesian optimization using information-based search. *The Journal of Machine Learning Research*, pages 5549–5601, 2016.

Torsten Koller, Felix Berkenkamp, Matteo Turchetta, and Andreas Krause. Learning-based model predictive control for safe exploration and reinforcement learning. In *Proc. of the IEEE Conference on Decision and Control (CDC)*, December 2018.

Alonso Marco, Felix Berkenkamp, Philipp Hennig, Angela P Schoellig, Andreas Krause, Stefan Schaal, and Sebastian Trimpe. Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with bayesian optimization. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 1557–1563. IEEE, 2017.

Alfred S McEwen, Eric M Eliason, James W Bergstrom, Nathan T Bridges, Candice J Hansen, W Alan Delamere, John A Grant, Virginia C Gulick, Kenneth E Herkenhoff, Laszlo Keszthelyi, et al. Mars reconnaissance orbiter's high resolution imaging science experiment (hirise). *Journal of Geophysical Research: Planets*, 112(E5), 2007.

Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. The application of bayesian methods for seeking the extremum. *Towards global optimization*, (117-129), 1978.

Teodor Mihai Moldovan and Pieter Abbeel. Safe exploration in markov decision processes. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pages 1451–1458, 2012.

MSL. MSL Landing Site Selection User's Guide to Engineering Constraints, 2007. URL http://marsoweb.nas.nasa.gov/landingsites/msl/memoranda/MSL_Eng_User_Guide_v4.5.1.pdf.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Carl Edward Rasmussen. Gaussian processes in machine learning. In *Advanced lectures on machine learning*, pages 63–71. Springer, 2004.

Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

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.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 1015–1022, 2010.

Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

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, 2015.

Yanan Sui, Vincent Zhuang, Joel W Burdick, and Yisong Yue. Stagewise safe bayesian optimization with gaussian processes. *arXiv preprint arXiv:1806.07555*, 2018.

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

Matteo Turchetta, Felix Berkenkamp, and Andreas Krause. Safe exploration in finite markov decision processes with gaussian processes. In *Advances in Neural Information Processing Systems*, pages 4312–4320, 2016.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake Vand erPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1. 0 Contributors. SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python. *arXiv e-prints*, art. arXiv:1907.10121, Jul 2019.

Akifumi Wachi, Yanan Sui, Yisong Yue, and Masahiro Ono. Safe exploration and optimization of constrained mdps using gaussian processes. *Association for the Advancement of Artificial Intelligence*, 2018.

Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient bayesian optimization. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3627–3635. JMLR. org, 2017.

Douglas J White. A survey of applications of markov decision processes. *Journal of the operational research society*, pages 1073–1096, 1993.

Yifan Wu, Roshan Shariff, Tor Lattimore, and Csaba Szepesvári. Conservative bandits. In *International Conference on Machine Learning*, pages 1254–1262, 2016.

In the following we present the proof of our result.

A Definitions

For ease of consultation we repeat the relevant definitions here. We denote with $\mathcal{G} = (\mathcal{D}, E)$ the directed graph describing the dependency among decisions introduced in Sec. 2, where $\mathcal{D}$ indicates the vertices of the graph, i.e., the decision space of the problem and $E \subseteq \mathcal{D} \times \mathcal{D}$ denotes the edges. Baseline for safety:

$$R_\epsilon^{\text{safe}}(S) = S \cup \{\mathbf{x} \in \mathcal{D} \setminus S, | \exists \mathbf{z} \in S \colon q(\mathbf{z}) - \epsilon - Ld(\mathbf{x}, \mathbf{z}) \geq 0\}, \tag{3}$$

The ergodicity operator is defined by intersecting the nodes that are reachable from a set S and the nodes from which we can return to a set $\overline{S}$ through a path contained in another set S:

$$R^{\text{reach}}(S) = S \cup \{\mathbf{x} \in \mathcal{D} \setminus S \,|\, \exists \mathbf{z} \in S : (\mathbf{z}, \mathbf{x}) \in E\}, \tag{4}$$

$$R_n^{\text{reach}}(S) = R^{\text{reach}}(R_{n-1}^{\text{reach}}(S)) \text{ with } R_1^{\text{reach}}(S) = R^{\text{reach}}(S), \tag{5}$$

$$\tilde{R}^{\text{reach}}(S) = \lim_{n \to \infty} R_n^{\text{reach}}(S), \tag{6}$$

$$R^{\text{ret}}(S, \overline{S}) = \overline{S} \cup \{\mathbf{x} \in S \,|\, \exists \mathbf{z} \in \overline{S} : (\mathbf{x}, \mathbf{z}) \in E\} \tag{7}$$

$$R_n^{\text{ret}}(S, \overline{S}) = R^{\text{ret}}(S, R_{n-1}^{\text{ret}}(S, \overline{S})), \text{ with } R_1^{\text{ret}}(S, \overline{S}) = R^{\text{ret}}(S, \overline{S}), \tag{8}$$

$$\tilde{R}^{\text{ret}}(S, \overline{S}) = \lim_{n \to \infty} R_n^{\text{ret}}(S, \overline{S}), \tag{9}$$

$$R^{\text{ergodic}}(S, \overline{S}) = \tilde{R}^{\text{reach}}(\overline{S}) \cap \tilde{R}^{\text{ret}}(S, \overline{S}). \tag{10}$$

Here, we repeat the definition of the safe and ergodic baseline introduced by Turchetta et al. (2016):

$$R_\epsilon(S) = R_\epsilon^{\text{safe}}(S) \cap R^{\text{ergodic}}(R_\epsilon^{\text{safe}}(S), S), \tag{11}$$

$$R_\epsilon^n(S) = R_\epsilon(R_\epsilon^{n-1}(S)) \text{ with } R_\epsilon^1 = R_\epsilon(S), \tag{12}$$

$$\overline{R}_\epsilon(S) = \lim_{n \to \infty} R_\epsilon^n(S). \tag{13}$$

Optimistic and pessimistic constraint satisfaction operators:

$$o_t^\epsilon(S) = \{\mathbf{x} \in \mathcal{D}, \,|\, \exists \mathbf{z} \in S : u_t(\mathbf{z}) - Ld(\mathbf{x}, \mathbf{z}) - \epsilon \geq 0\}, \tag{14}$$

$$p_t^\epsilon(S) = \{\mathbf{x} \in \mathcal{D}, \,|\, \exists \mathbf{z} \in S : l_t(\mathbf{z}) - Ld(\mathbf{x}, \mathbf{z}) - \epsilon \geq 0\} \tag{15}$$

Optimistic expansion operator:

$$O_t^{\epsilon,1}(S) = o_t^\epsilon(S) \cap R^{\text{ergodic}}(o_t^\epsilon(S), S), \tag{16}$$

$$O_t^{\epsilon,n}(S) = o_t^\epsilon(O_t^{\epsilon,n-1}(S)) \cap R^{\text{ergodic}}(o_t^\epsilon(O_t^{\epsilon,n-1}(S)), S), \tag{17}$$

$$\tilde{O}_t^\epsilon(S) = \lim_{n \to \infty} O_t^{n,\epsilon}(S) \tag{18}$$

Pessimistic expansion operators:

$$P_t^{\epsilon,1}(S) = p_t^\epsilon(S) \cap R^{\text{ergodic}}(p_t^\epsilon(S), S), \tag{19}$$

$$P_t^{\epsilon,n}(S) = p_t^\epsilon(P_t^{\epsilon,n-1}(S)) \cap R^{\text{ergodic}}(p_t^\epsilon(P_t^{\epsilon,n-1}(S)), S), \tag{20}$$

$$\tilde{P}_t^\epsilon(S) = \lim_{n \to \infty} P_t^{n,\epsilon}(S) \tag{21}$$

Pessimistic and optimistic safe and ergodic sets:

$$\bar{S}_t^{o,\epsilon} = \tilde{O}_t^\epsilon(\bar{S}_{t-1}^p) \tag{22}$$

$$\bar{S}_t^p = \tilde{P}_t^0(\bar{S}_{t-1}^p), \tag{23}$$

Points with uncertainty above threshold:

$$W_t^\epsilon = \{\mathbf{x} \in \bar{S}_t^p : w_t(\mathbf{x}) > \epsilon\}. \tag{24}$$

Set of decisions with equal priority:

$$A_t(\alpha) = \{\mathbf{x} \in \bar{S}_t^{o,\epsilon} \setminus p_t^0(, \bar{S}_t^p) : h_t(\mathbf{x}) = \alpha\} \tag{25}$$

Immediate expanders for nodes with priority α:

$$G_t^\epsilon(\alpha) = \{\mathbf{x} \in W_t^\epsilon, \,|\, \exists \mathbf{z} \in A_t(\alpha) \colon u_t(\mathbf{x}) - Ld(\mathbf{x}, \mathbf{z}) \geq 0\}., \tag{26}$$

Relevant priority:

$$\alpha^* = \max \ \alpha, \ \text{s.t.} \ |G_t^\epsilon(\alpha)| > 0. \tag{27}$$

Notice that our definition of $\bar{S}_t^p$ differs slightly from the one in Turchetta et al. (2016) in that we alternate the pessimistic expansion step and the restriction to ergodic nodes until convergence, whereas Turchetta et al. (2016) do it only once at each time step. In particular, we have $\bar{S}_t^p = \lim_{n\to\infty} P_t^{0,n}(\bar{S}_{t-1}^p)$, while they use $\bar{S}_t^p = P_t^{0,1}(\bar{S}_{t-1}^p)$. In practice, this does not make any difference since it is easy to verify that, by t^*, i.e., the time by when both GoOSE and the approach in Turchetta et al. (2016) are guaranteed to converge in the worst case, the pessimistic ergodic safe sets should be the same for both methods. However, our new definition allows for a more efficient exploration. These new definitions would require us to show again some of the properties that were shown by Turchetta et al. (2016) for $\bar{S}^p$. However, due to our recursive definition of $\bar{S}^p$, it is easy to see that it is possible to show these properties by induction over the index n. In this case, the lemmas introduced by Turchetta et al. (2016) constitute the base case. At this point, it is sufficient to use the induction hypothesis and the monotonicity of the confidence interval shown in Lemma 1 together with basic properties of the R^{ergodic} operator discussed in Lemma 4 to prove the induction step. We show how to do this explicitly in Lemma 5. However, we do not explicitly repeat this reasoning for every lemma involving $\bar{S}^p$ and we refer to Turchetta et al. (2016) instead.

B   Preliminary lemmas

This section contains some basic lemmas about the sets defined above that will be used in subsequent sections to prove our main results.

Lemma 1. $\forall \boldsymbol{x} \in \mathcal{D}, \ u_{t+1}(\boldsymbol{x}) \leq u_t(\boldsymbol{x}), \ l_{t+1}(\boldsymbol{x}) \geq l_t(\boldsymbol{x}), \ w_{t+1}(\boldsymbol{x}) \leq w_t(\boldsymbol{x}).$

Proof. See Lemma 1 in Turchetta et al. (2016). $\qquad\square$

Lemma 2. *Given* $S \subseteq R \subseteq \mathcal{D}$ *and* $\overline{S} \subseteq \overline{R} \subseteq \mathcal{D}$, *it holds that* $\tilde{R}^{\text{ret}}(\overline{S}, S) \subseteq \tilde{R}^{\text{ret}}(\overline{R}, R).$

Proof. See Lemma 7 in Turchetta et al. (2016) $\qquad\square$

Lemma 3. *For any* $S, R \subseteq \mathcal{D}$, *for any* $n \geq 1$ *we have* $R_n^{\text{reach}}(S) \subseteq R_n^{\text{reach}}(R).$

We proceed by induction. For $n = 1$, we have $R^{\text{reach}}(S) \subseteq R^{\text{reach}}(R)$ by Lemma 8 by Turchetta et al. (2016). For the inductive step, assume $R_{n-1}^{\text{reach}}(S) \subseteq R_{n-1}^{\text{reach}}(R)$. Consider $\mathbf{x} \in R_n^{\text{reach}}(S)$. We know $\exists \mathbf{x}' \in R_{n-1}^{\text{reach}}(S) \subseteq R_{n-1}^{\text{reach}}(R)$, $a \in \mathcal{A}(\mathbf{x}')$ such that $\mathbf{x} = f(\mathbf{x}', a)$, which implies $\mathbf{x} \in R_n^{\text{reach}}(R)$.

Corollary 2. *For any* $S, R \subseteq \mathcal{D}$, *we have* $\tilde{R}^{\text{reach}}(S) \subseteq \tilde{R}^{\text{reach}}(R).$

Lemma 4. *For any* $S, R \subseteq \mathcal{D}$ *and* $\overline{S}, \overline{R} \subseteq \mathcal{D}$, *we have* $R^{\text{ergodic}}(S, \overline{S}) \subseteq R^{\text{ergodic}}(R, \overline{R}).$

Proof. This follows from Lemma 2 and Corollary 2. $\qquad\square$

Lemma 5. *For any* $t \geq 1$, $\bar{S}_0^p \subseteq \bar{S}_t^p \subseteq \bar{S}_{t+1}^p.$

Proof. Lemma 9 in Turchetta et al. (2016) allows us to say $P_t^{0,1}(\bar{S}_{t-1}^p) \subseteq P_{t+1}^{0,1}(\bar{S}_t^p)$. Thus, we can assume $P_t^{0,n-1}(\bar{S}_{t-1}^p) \subseteq P_{t+1}^{0,n-1}(\bar{S}_t^p)$ as induction hypothesis. Let us consider $\mathbf{x} \in P_t^{0,n-1}(\bar{S}_{t-1}^p)$. We know there exists $\mathbf{z} \in P_t^{0,n-1}(\bar{S}_{t-1}^p) \subseteq P_{t+1}^{0,n-1}(\bar{S}_t^p)$ such that $l_t(\mathbf{z}) - Ld(\mathbf{x}, \mathbf{z}) \geq 0$, which ,by Lemma 1, implies $l_{t+1}(\mathbf{z}) - Ld(\mathbf{x}, \mathbf{z}) \geq 0$. This means that $p_t^0(P_t^{0,n-1}(\bar{S}_{t-1}^p)) \subseteq p_{t+1}^0(P_{t+1}^{0,n-1}(\bar{S}_t^p))$. Applying Lemma 4, we complete the induction step and show $P_t^{0,n}(\bar{S}_{t-1}^p) \subseteq P_{t+1}^{0,n}(\bar{S}_t^p)$. $\qquad\square$

Lemma 6. *(Chowdhury and Gopalan, 2017, Thm. 2) Assume* $\|q\|_k^2 \leq B_q$, *and* σ-*sub-Gaussian noise. If* $\beta_t^{1/2} = B_q + 4\sigma\sqrt{\gamma_t + 1 + \ln(1/\delta)}$, *then, for all* $t > 0$ *and all* $\boldsymbol{x} \in \mathcal{D}$, $|q(\boldsymbol{x}) - \mu_{t-1}(\boldsymbol{x})| \leq \beta_t^{1/2}\sigma_{t-1}(\boldsymbol{x})$ *holds with probability at least* $1 - \delta$.

Proof. See Theorem 2 in Chowdhury and Gopalan (2017). □

Lemma 7. *Let* $\beta_t^{1/2} = B_q + 4\sigma\sqrt{\gamma_t + 1 + \ln(1/\delta)}$ *and assume* $\|q\|_k^2 \leq B_q$, *and* σ-*sub-Gaussian noise. Then, for all* $t > 0$ *and all* $\boldsymbol{x} \in \mathcal{D}$, *it holds with probability at least* $1 - \delta$ *that* $q(\boldsymbol{x}) \in C_t(\boldsymbol{x})$.

Proof. See Corollary 1 in Sui et al. (2015). □

C   Safety

The safety of our algorithm depends on the confidence intervals and on the safe and ergodic set $\bar{S}_t^p$. Since these are defined as in Turchetta et al. (2016), their safety guarantees carry over to our case.

Theorem 2. *For any node* $\boldsymbol{x}$ *along any trajectory induced by Alg. 1 on the graph* $\mathcal{G}$ *we have, with probability at least* $1 - \delta$, *that* $q(\boldsymbol{x}) \geq 0$.

Proof. See Theorem 2 in Turchetta et al. (2016). □

D   Completeness

In this section, we develop the core of our theoretical contribution. The analysis in Turchetta et al. (2016) bounds the uncertainty of the expanders when the safe set does not change in an interval $[t_0, t_1]$ without considering the measurements collected prior to t_0. By considering this information, we extend their worst case sample complexity bound to our more general formulation of the safe exploration problem.

The following lemmas refer to the exploration steps, i.e., when the goal suggested by the oracle $\mathcal{O}$ lies outside of the pessimistic safe and ergodic set (Line 5, Alg. 1). Notice that t denotes the number of constraint evaluations and it differs from the iteration index of the algorithm i.

The core idea is the following: We bound the number constraint evaluations required at point in the domain to guarantee that its uncertainty is below ϵ. We show that, as a consequence, if the safe and ergodic set does not change for long enough all the expanders have uncertainty below ϵ. At this point we can either guarantee that the safe set expands or that the whole $\tilde{R}_\epsilon(S_0)$ has been explored. Since the analysis relies on the number of constraint evaluations at each point in the domain, we can evaluate them in any order as long as we exclude those that have an uncertainty below ϵ. Therefore, our exploration guarantees hold for any priority function.

In the following, let us denote with $\mathcal{T}_t^{\mathbf{x}} = \{\tau_1, \cdots, \tau_j\}$ the set of steps where the constraint q is evaluated at $\mathbf{x}$ by step t. Moreover, we assume, without loss of generality, $k(\mathbf{x}, \mathbf{x}) \leq 1$, i.e., we assume bounded variance.

Lemma 8. *For any* $t \geq 1$ *and for any* $\boldsymbol{x} \in \mathcal{D}$, *it holds that* $w_t(\boldsymbol{x}) \leq \sqrt{\frac{C_1 \gamma_t \beta_t}{|\mathcal{T}_t^{\mathbf{x}}|}}$, *with* $C_1 = 8/\log(1 - \sigma^{-2})$.

Proof.

$$|\mathcal{T}_t^{\mathbf{x}}| w_t^2(\mathbf{x}) \leq \sum_{\tau \in \mathcal{T}_t^{\mathbf{x}}} w_\tau^2(\mathbf{x}) \tag{28}$$

$$\leq \sum_{\tau \in \mathcal{T}_t^{\mathbf{x}}} 4\beta_\tau \sigma_{\tau-1}^2(\mathbf{x}), \tag{29}$$

$$\leq \sum_{\tau \leq t} 4\beta_\tau \sigma_{\tau-1}^2(\mathbf{x}), \tag{30}$$

$$\leq C_1 \gamma_t \beta_t, \tag{31}$$

with $C_1 = 8/\log(1 - \sigma^{-2})$. Here, (28) holds because of Lemma 1, and (31) holds because of Lemma 5.4 by Srinivas et al. (2010). □

For the remainder of the paper, let us denote with T_t the smallest positive integer such that $\frac{T_t}{\beta_{t+T_t}\gamma_{t+T_t}} \geq \frac{C_1}{\epsilon^2}$, with $C_1 = 8/\log(1 - \sigma^{-2})$ and with t^* the smallest positive integer such that $t^* \geq |\tilde{R}_0(S_0)|T_{t^*}$.

Lemma 9. *For any $t \leq t^*$, for any $\boldsymbol{x} \in \mathcal{D}$ such that $|\mathcal{T}_t^{\boldsymbol{x}}| \geq T_{t^*}$, it holds that $w_t(\boldsymbol{x}) \leq \epsilon$.*

Proof. Since T_t is an increasing function of t (Sui et al., 2015), we have $|\mathcal{T}_t^{\boldsymbol{x}}| \geq T_{t^*} \geq T_t$. Therefore, using Lemma 8 and the definition of T_t, we have

$$w_t(\mathbf{x}) \leq \sqrt{\frac{C_1\gamma_t\beta_t}{T_t}} \leq \sqrt{\frac{C_1\gamma_t\beta_t\epsilon^2}{C_1\gamma_{t+T_t}\gamma_{t+T_t}}} \leq \sqrt{\frac{\gamma_t\beta_t}{\gamma_{t+T_t}\gamma_{t+T_t}}}\,\epsilon \leq \epsilon, \tag{32}$$

where the last inequality comes from the fact that both β_t and γ_t are increasing functions of t. $\square$

Lemma 10. *For any $t \leq t^*$, $|\mathcal{T}_t^{\boldsymbol{x}}| \leq T_{t^*}$, for any $\boldsymbol{x} \in \bar{S}_t^p$.*

Proof. According to Line 6 of Alg. 2, we only evaluate the constraint at points $\mathbf{x} \in \mathcal{D}$ if $w_t(\mathbf{x}) > \epsilon$. From Lemma 9 we know that $|\mathcal{T}_t^{\mathbf{x}}| = T_{t^*} \implies w_t(\mathbf{x}) \leq \epsilon$. Thus, if $|\mathcal{T}_t^{\mathbf{x}}| = T_{t^*}$, $\mathbf{x}$ is not evaluated anymore, which means that $|\mathcal{T}_t^{\mathbf{x}}|$ cannot grow any further. $\square$

Lemma 11. *$\forall t \geq 0$, $\bar{S}_t^p \subseteq \overline{R}_0(S_0)$ with probability at least $1 - \delta$.*

Proof. See Lemma 22 in Turchetta et al. (2016). $\square$

The following lemma bounds the uncertainty of the points sampled by GOOSE when the set of safe and ergodic points does not increase.

Lemma 12. *For any $t \leq t^*$, let $\tau_t = |\bar{S}_t^p|T_{t^*}$, if $\bar{S}_t^p = \bar{S}_{\tau_t}^p$, then $w_{\tau_t}(\boldsymbol{x}) \leq \epsilon$ for all $\boldsymbol{x} \in \cup_\alpha G_{\tau_t}^\epsilon(\alpha)$.*

Proof. First, we notice that

$$\sum_{\mathbf{x}\in\bar{S}_{\tau_t}^p} |\mathcal{T}_{\tau_t}^{\mathbf{x}}| = \tau_t = |\bar{S}_t^p|T_{t^*} = |\bar{S}_{\tau_t}^p|T_{t^*}, \tag{33}$$

where the first equality comes from the fact that the sum of the number of constraint observations in τ_t time steps is equal to τ_t, the second comes from the definition of τ_t and the third comes from the assumption that $\bar{S}_t^p = \bar{S}_{\tau_t}^p$. This allows us to say that, for all $\mathbf{x} \in \bar{S}_{\tau_t}^p$

$$\sum_{\mathbf{z}\in\bar{S}_{\tau_t}^p\setminus\{\mathbf{x}\}} |\mathcal{T}_{\tau_t}^{\mathbf{z}}| = |\bar{S}_{\tau_t}^p|T_{t^*} - |\mathcal{T}_{\tau_t}^{\mathbf{x}}|. \tag{34}$$

Moreover, we have $\tau_t = |\bar{S}_t^p|T_{t^*} \leq |\tilde{R}_0(S_0)|T_{t^*} \leq t^*$ by definition of t^* and τ_t and Lemma 11. Therefore, by Lemma 10 we know that $T_{t^*} \geq |\mathcal{T}_{\tau_t}^{\mathbf{x}}|$ for all $\mathbf{x} \in \bar{S}_{\tau_t}^p$. Now we show by contradiction that $T_{t^*} = |\mathcal{T}_{\tau_t}^{\mathbf{x}}|$ for all $\mathbf{x} \in \bar{S}_{\tau_t}^p$. Assume this is not the case, i.e., there is $\mathbf{x} \in \bar{S}_{\tau_t}^p$ such that $T_{t^*} > |\mathcal{T}_{\tau_t}^{\mathbf{x}}|$. We have

$$(|\bar{S}_{\tau_t}^p| - 1)T_{t^*} \geq \sum_{\mathbf{z}\in\bar{S}_{\tau_t}^p\setminus\{\mathbf{x}\}} |\mathcal{T}_{\tau_t}^{\mathbf{z}}| = |\bar{S}_{\tau_t}^p|T_{t^*} - |\mathcal{T}_{\tau_t}^{\mathbf{x}}| > |\bar{S}_{\tau_t}^p|T_{t^*} - T_{t^*} = (|\bar{S}_{\tau_t}^p| - 1)T_{t^*}, \tag{35}$$

which is a contradiction and proves our claim that $T_{t^*} = |\mathcal{T}_{\tau_t}^{\mathbf{x}}|$ for all $\mathbf{x} \in \bar{S}_{\tau_t}^p$. Therefore, by Lemma 9, $w_{\tau_t}(\mathbf{x}) \leq \epsilon$ for all $\mathbf{x} \in \bar{S}_{\tau_t}^p$. This proves our claim since $\cup_\alpha G_{\tau_t}^\epsilon(\alpha) \subseteq \bar{S}_{\tau_t}^p$. $\square$

Lemma 13. *For any $t \geq 1$, $\overline{R}_\epsilon(S_0) \setminus \bar{S}_t^p \neq \emptyset$, then, $R_\epsilon(\bar{S}_t^p) \setminus \bar{S}_t^p \neq \emptyset$.*

Proof. See Lemma 20 in Turchetta et al. (2016). $\square$

Lemma 14. *For any $t \leq t^*$, if $\tilde{R}_\epsilon(S_0) \setminus \bar{S}_t^p \neq \emptyset$, then $\bar{S}_t^p \subset \bar{S}_{|\hat{S}_t|T_{t^*}}^p$ with probability at least $1 - \delta$.*

Proof. This proof is analogous to Lemma 21 in Turchetta et al. (2016) where we use our Lemma 12 rather than their Lemma 19 to bound the uncertainty of the expanders. $\square$

15

Lemma 15. *There exists $t \leq t^*$ such that $\tilde{R}_\epsilon(S_0) \subseteq \bar{S}_t^p$ with probability at least $1 - \delta$.*

Proof. For the sake of contradiction, assume this is not the case and that $\forall t \leq t^*$ holds that $\tilde{R}_\epsilon(S_0) \setminus \bar{S}_t^p \neq \emptyset$. For all $i \geq 1$ define $\tau_i = |\bar{S}_{\tau_{i-1}}^p| T_{t^*}$ with $\tau_0 = 0$. We know that $\tau_i \leq t^*$ for all i because of Lemma 11 and that $\tau_0 \leq \tau_1 \leq \cdots$ because of Lemma 5. Therefore, Lemma 14 implies that $\bar{S}_0^p \subset \bar{S}_{\tau_1}^p \subset \bar{S}_{\tau_2}^p \subset \cdots$. In general, this means that $|\bar{S}_{\tau_i}^p| \geq |\bar{S}_0^p| + i$ for all $i \geq 1$. In particular if we set $i = |\tilde{R}_0(S_0) \setminus S_0| + 1$ we get that $|\bar{S}_{\tau_i}^p| \geq |\bar{S}_0^p| + |\tilde{R}_0(S_0) \setminus S_0| + 1 = |\tilde{R}_0(S_0)| + 1 > |\tilde{R}_0(S_0)|$. This is a contradiction because of Lemma 11. $\square$

Lemma 16. *There is $t \leq t^*$ such that $\tilde{R}_\epsilon(S_0) \subseteq \bar{S}_t^p \subseteq \overline{R}_0(S_0)$ with probability at least $1 - \delta$.*

Proof. The lemma follows directly from Lemmas 11 and 15. $\square$

Lemma 17. $\bar{S}_{t^*}^{o,\epsilon} \subseteq \bar{S}_{t^*}^p$.

Proof. Since the optimistic and ergodic and safe set is defined recursively, we will prove this claim by induction. Similarly to Lemma 12, we start by noticing that, for every $\mathbf{x} \in \bar{S}_{t^*}^p$, we have:

$$\sum_{\mathbf{x} \in \bar{S}_{t^*}^p} |\mathcal{T}_{t^*}^{\mathbf{x}}| = t^* \geq |\tilde{R}_0(S_0)| T_{t^*}, \tag{36}$$

$$\implies \sum_{\mathbf{z} \in \bar{S}_{t^*}^p \setminus \mathbf{x}} |\mathcal{T}_{t^*}^{\mathbf{z}}| \geq |\tilde{R}_0(S_0)| T_{t^*} - |\mathcal{T}_{t^*}^{\mathbf{x}}| \geq |\bar{S}_{t^*}^p| T_{t^*} - |\mathcal{T}_{t^*}^{\mathbf{x}}|. \tag{37}$$

Lemma 10 allows us to say $|\mathcal{T}_{t^*}^{\mathbf{x}}| \leq T_{t^*}$ for all $\mathbf{x} \in \bar{S}_{t^*}^p$. We show by contradiction that $|\mathcal{T}_{t^*}^{\mathbf{x}}| = T_{t^*}$ for all $\mathbf{x} \in \bar{S}_{t^*}^p$. Assume this is not the case and that we have $\mathbf{x} \in \bar{S}_{t^*}^p$ such that $|\mathcal{T}_{t^*}^{\mathbf{x}}| < T_{t^*}$:

$$(|\bar{S}_{t^*}^p| - 1) T_{t^*} \geq \sum_{\mathbf{z} \in \bar{S}_{t^*}^p \setminus \{\mathbf{x}\}} |\mathcal{T}_{t^*}^{\mathbf{z}}| \geq |\bar{S}_{t^*}^p| T_{t^*} - |\mathcal{T}_{t^*}^{\mathbf{x}}| > |\bar{S}_{t^*}^p| T_{t^*} - T_{t^*} = (|\bar{S}_{t^*}^p| - 1) T_{t^*}, \tag{38}$$

which is a contradiction and proves that $|\mathcal{T}_{t^*}^{\mathbf{x}}| = T_{t^*}$ for all $\mathbf{x} \in \bar{S}_{t^*}^p$. Therefore, by Lemma 9, we know that $w_{t^*}(\mathbf{x}) \leq \epsilon$ for any $\mathbf{x} \in \bar{S}_{t^*}^p$. Now consider $\mathbf{x} \in o_{t^*}^\epsilon(\bar{S}_{t^*-1}^p)$. We know that there is $\mathbf{z} \in \bar{S}_{t^*-1}^p \subseteq \bar{S}_{t^*}^p$ such that $u_{t^*}(\mathbf{z}) - Ld(\mathbf{x}, \mathbf{z}) - \epsilon \geq 0$. Since $w_{t^*}(\mathbf{z}) \leq \epsilon$, we know that $l_{t^*}(\mathbf{z}) - Ld(\mathbf{x}, \mathbf{z}) \geq 0$, i.e., $\mathbf{x} \in p_{t^*}^0(\bar{S}_{t^*-1}^p)$. Using Lemma 4, we can say $O_{t^*}^{\epsilon,1}(\bar{S}_{t^*-1}^p) \subseteq P_{t^*}^{0,1}(\bar{S}_{t^*-1}^p)$. Now, we can make the following induction hypothesis: $O_{t^*}^{\epsilon,n-1}(\bar{S}_{t^*-1}^p) \subseteq P_{t^*}^{0,n-1}(\bar{S}_{t^*-1}^p)$. Consider $\mathbf{x} \in O_{t^*}^\epsilon(\bar{S}_{t^*-1}^p)$, we know there is $\mathbf{z} \in O_{t^*}^{\epsilon,n-1}(\bar{S}_{t^*-1}^p) \subseteq P_{t^*}^{0,n-1}(\bar{S}_{t^*-1}^p) \subseteq \bar{S}_{t^*}^p$ (where the first inclusion comes from the induction hypothesis and the second by definition of the safe set), such that $u_{t^*}(\mathbf{z}) - Ld(\mathbf{x}, \mathbf{z}) - \epsilon \geq 0$. Since $w_{t^*}(\mathbf{z}) \leq \epsilon$, we know that $l_{t^*}(\mathbf{z}) - Ld(\mathbf{x}, \mathbf{z}) \geq 0$, i.e., $\mathbf{x} \in p_{t^*}^0(P_{t^*}^{0,n-1}(\bar{S}_{t^*-1}^p))$. We can apply Lemma 4 again to complete the induction step and show $O_{t^*}^{\epsilon,n}(\bar{S}_{t^*-1}^p) \subseteq P_{t^*}^{0,n}(\bar{S}_{t^*-1}^p)$ and, therefore, $\bar{S}_{t^*}^{o,\epsilon} \subseteq \bar{S}_{t^*}^p$. $\square$

Lemma 18. *For every $t \geq 0$, we have $\tilde{R}_\epsilon(S_0) \subseteq \bar{S}_t^{o,\epsilon}$.*

Proof. We will show this claim with a proof by induction. Let us consider $\mathbf{x} \in R_\epsilon(S_0)$. We know there is a $\mathbf{z} \in S_0$ such that $q(\mathbf{z}) - Ld(\mathbf{x}, \mathbf{z}) - \epsilon \geq 0$. By Lemma 7, we know this means $u_t(\mathbf{z}) - Ld(\mathbf{x}, \mathbf{z}) - \epsilon \geq 0$ for all $t \geq 0$. Therefore, $R_\epsilon^{\text{safe}}(S_0) \subseteq o_t^\epsilon(S_0) \subseteq o_t^\epsilon(\bar{S}_{t-1}^p)$ since $S_0 \subseteq \bar{S}_{t-1}^p$ for all $t \geq 1$ by Lemma 5. Lemma 4 allows us to say that $R_\epsilon(S_0) \subseteq O_t^\epsilon(\bar{S}_{t-1}^p)$. As induction hypothesis, we can assume $R_\epsilon^{n-1}(S_0) \subseteq O_t^{\epsilon,n-1}(\bar{S}_{t-1}^p)$. Consider $\mathbf{x} \in R_\epsilon^n(S_0)$. We know there is $\mathbf{z} \in R_\epsilon^{n-1}(S_0) \subseteq O_t^{\epsilon,n-1}(\bar{S}_{t-1}^p)$ such that $q(\mathbf{z}) - Ld(\mathbf{x}, \mathbf{z}) - \epsilon \geq 0$ which, by Lemma 7, means that $u_t(\mathbf{z}) - Ld(\mathbf{x}, \mathbf{z}) - \epsilon \geq 0$ for all $t \geq 0$. Therefore, $R_\epsilon^{\text{safe}}(R_\epsilon^{n-1}(S_0)) \subseteq o_t^\epsilon(O_t^{\epsilon,n-1}(\bar{S}_{t-1}^p))$. Using Lemma 4, we can say $R_\epsilon^n(S_0) \subseteq O_t^{\epsilon,n}(\bar{S}_{t-1}^p)$, which completes the induction step and concludes the proof. $\square$

E Main result

Theorem 1. *Assume that $q(\cdot)$ is L-Lipschitz continuous w.r.t. $d(\cdot, \cdot)$ with $\|q\|_k \leq B_q$, σ-sub-Gaussian noise, $S_0 \neq \emptyset$, $q(\boldsymbol{x}) \geq 0$ for all $\boldsymbol{x} \in S_0$, and that, for any two decisions $\boldsymbol{x}, \boldsymbol{x}' \in S_0$, there is a path in the graph $\mathcal{G}$ connecting them within S_0. Let $\beta_t^{1/2} = B_q + 4\sigma\sqrt{\gamma_t + 1 + \ln(1/\delta)}$, then, for any $h_t : \mathcal{D} \to \mathbb{R}$, with probability at least $1 - \delta$, we have $q(\boldsymbol{x}) \geq 0$ for any $\boldsymbol{x}$ visited by GoOSE. Moreover, let γ_t denote the information capacity associated with the kernel k and let t^* be the smallest integer such that $\frac{t^*}{\beta_{t^*}\gamma_{t^*}} \geq \frac{C|\tilde{R}_0(S_0)|}{\epsilon^2}$, with $C = 8/\log(1 + \sigma^{-2})$, then there exists a $t \leq t^*$ such that, with probability at least $1 - \delta$, $\tilde{R}_\epsilon(S_0) \subseteq \bar{S}_t^{o,\epsilon} \subseteq \bar{S}_t^p \subseteq \tilde{R}_0(S_0)$.*

Proof. The safety is a direct consequence of Theorem 2. The convergence of the pessimistic and optimistic approximation of the safe sets is a direct consequence of Lemmas 16–18. □

The following corollary gives a simpler interpretation of our main result: in presence of an unknown constraint, the IML oracle augmented with GoOSE behaves as the IML oracle would behave if it had knowledge of a better-than-ϵ-accurate approximation of the safe reachable set from the beginning (except for a finite number of constraint evaluations).

Corollary 1. *Under the assumptions of Theorem 1, let the IML oracle be deterministic given the observations. Then there exists a set S with $\tilde{R}_\epsilon(S_0) \subseteq S \subseteq \tilde{R}_0(S_0)$ so that $\boldsymbol{x}_i^\star = \mathcal{O}(S)$ for all $k \geq 1$.*

Proof. This is a direct consequence of Theorem 1, since it guarantees that, if necessary, we can expand the set of points where we can evaluate the objective (i.e. the pessimistic safe set) and we can contract the decision space of the IML oracle (i.e. the optimistic safe set) to a point where the first contains the second, in a finite number of constraint evaluations. □