Safe exploration in reproducing kernel Hilbert spaces

Abdullah Tokmak* **Kiran G. Krishnan*** **Thomas B. Schön†** **Dominik Baumann*†**

* Cyber-physical Systems Group
Aalto University, Espoo, Finland
abdullah.tokmak@aalto.fi

† Department of Information Technology
Uppsala University, Uppsala, Sweden

Abstract

Popular safe Bayesian optimization (BO) algorithms successfully control safety-critical systems in unknown environments. However, most algorithms require smoothness assumptions, which are encoded by a norm in a reproducing kernel Hilbert space (RKHS). The RKHS is a potentially infinite-dimensional space and it remains unclear how to reliably obtain the RKHS norm of an unknown function. In this work, we propose a safe BO algorithm capable of estimating the RKHS norm from data. We provide statistical guarantees on the RKHS norm estimation, derive novel confidence intervals for, and prove safety of the resulting safe BO algorithm. We apply our algorithm to safely optimize reinforcement learning policies on physics simulators and on a real Furuta pendulum, demonstrating improved performance, safety, and scalability compared to the state-of-the-art.

Keywords Safe Bayesian optimization, reproducing kernel Hilbert spaces, PAC learning, robotics.

1 Introduction

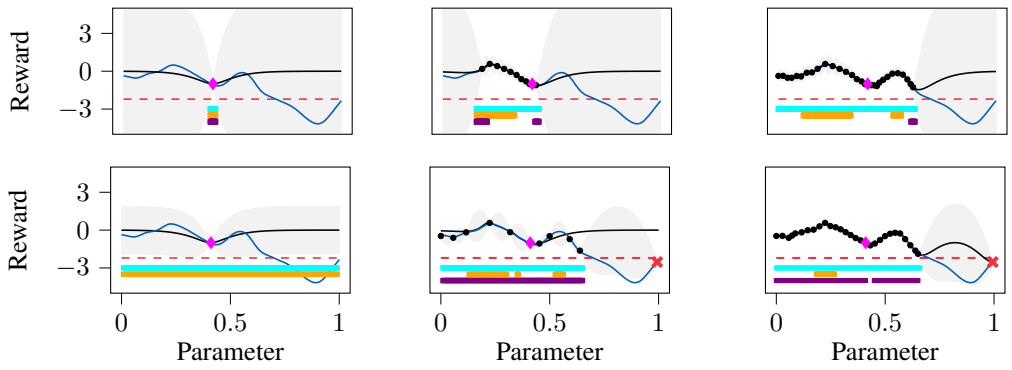

Figure 1: *Toy example of safe BO and the influence of the RKHS norm.* We aim to maximize the reward function (blue) while only sampling above the safety threshold (red dashed line). The predicted function (black line) is computed based on iteratively acquired samples (black dots, initial sample shown by the magenta diamond) and the confidence intervals are shown by the gray shaded area. At each iteration, we compute a set of parameters that we believe to be safe (cyan), potential expanders (purple), and potential maximizers (orange), thus safely balancing exploration and exploitation. The upper sub-figures show safe BO, where the true RKHS norm is used to compute the confidence intervals, while the lower sub-figures are generated with an under-estimated RKHS norm. An under-estimation of the RKHS norm can yield confidence intervals that do not contain the reward function, which may eventually lead to unsafe experiments (red cross).

38th Workshop on Aligning Reinforcement Learning Experimentalists and Theorists (ARLET 2024).

When learning policies for systems that act in the real world, such as mobile robots or autonomous vehicles, two crucial requirements must be met: *(i)* the learning algorithms we use must be sample efficient, as learning experiments are time-consuming and cause wear and tear to the hardware; and *(ii)* we must guarantee safety during exploration, i.e., while testing new policies, for systems not to damage themselves, their environment, or endanger people. Currently, one of the most popular tools for policy learning is reinforcement learning (RL). Without the need for a dynamics model, RL learns a policy through trial-and-error, i.e., by performing experiments and receiving a reward signal in return that it tries to maximize. Unfortunately, RL struggles with both requirements. Hence, the most impressive results of RL algorithms have been achieved in simulated or gaming environments [1–3].

An alternative to RL for policy learning is combining Bayesian optimization (BO) [4, 5] with Gaussian process (GP) [6] regression. When modeling the reward function with a GP, we can leverage this model and pose the decision of where to explore next as an optimization problem. This way of sequential decision-making dramatically improves sample efficiency, as shown in numerous hardware experiments [7–9]. Thus, combining GPs and BO meets the first requirement. For the second requirement, safe BO algorithms guarantee safety during exploration with high probability; a well-known example is SAFEOPT [10]. SAFEOPT, as well as other popular safe BO algorithms, assume that the reward function lies in a reproducing kernel Hilbert space (RKHS). However, guaranteeing safety requires an additional smoothness assumption, which is encoded by knowing a tight upper bound on the norm of the reward function in that RKHS. Even though the assumption elegantly paves the way to guarantee safety with high probability [11, Theorem 1], it is highly unrealistic since the RKHS is a potentially infinite-dimensional space, and it is unclear how to guess that upper bound for systems with unknown dynamics. If we misspecify the RKHS norm, i.e., if the true RKHS norm is higher than the bound we assume, safety guarantees become obsolete, as we illustrate in Figure 1.

Contribution. In response, we present a data-driven approach to compute an RKHS norm over-estimation with statistical guarantees. We integrate the RKHS norm over-estimation into a safe BO algorithm reminiscent of SAFEOPT, for which we derive novel confidence intervals and prove safety with high probability. Moreover, we extend our proposed safe BO algorithm by introducing a notion of locality. By considering local RKHS norms, which are potentially smaller than the global RKHS norm, we can explore more optimistically and improve scalability by separately discretizing local sub-domains. We compare our algorithm to SAFEOPT in a synthetic example and challenging robotic simulation benchmarks, where we demonstrate the benefits of over-estimating the RKHS norm from data instead of randomly guessing it. Finally, we demonstrate the applicability of our algorithm to real-world systems in a hardware experiment.

2 Problem setting and preliminaries

We cast safe policy search as a constrained optimization problem, where the objective function measures the performance. We consider parameterized policies, and the parameters, which could be the parameters of a controller, serve as the decision variables of the optimization problem.

Problem setting. We aim to maximize an unknown reward function $f\colon \mathcal{A} \subseteq \mathbb{R}^n \to \mathbb{R}$ while guaranteeing safety. We define safety as only sampling parameters $a \in \mathcal{A}$ that correspond to reward values larger than a pre-defined safety threshold $h \in \mathbb{R}$. Thus, we write the optimization problem as

$$\max_{a \in \mathcal{A}} f(a) \quad \text{subject to} \quad f(a) \geq h. \tag{1}$$

We solve (1) by sequentially querying the reward function at each iteration $t \in \mathbb{N}$. In return, we receive measurements $y_t \coloneqq f(a_t) + \epsilon_t$, where ϵ_t is independent and identically distributed (i.i.d.) σ-sub-Gaussian measurement noise. We denote the queried parametrizations until iteration t by $a_{1:t} \coloneqq [a_1, \ldots, a_\mathrm{t}]^\top$ and the corresponding measurements are denoted by $y_{1:t}$. GP regression provides a natural tool to estimate f, as done in SAFEOPT [10] and numerous other BO algorithms [11, 12]. Given data $a_{1:t}$ and $y_{1:t}$ at each iteration t, the posterior GP mean and covariance are

$$\begin{aligned} \mu_t(a) &= k_t(a)^\top (K_t + \sigma^2 I_t)^{-1} y_{1:t}, \\ \sigma_t^2(a) &= k(a,a) - k_t(a)^\top (K_t + \sigma^2 I_t)^{-1} k_t(a), \end{aligned} \tag{2}$$

respectively [6], where $k(a,a)$ is the kernel evaluated at $a \in \mathcal{A}$, $k_t(a) = [k(a,a_1), \ldots, k(a,a_\mathrm{t})]^\top \in \mathbb{R}^t$ the covariance vector, $K_t \in \mathbb{R}^{t \times t}$ the covariance matrix with entry $k(a_i, a_j)$ at row i and column j for all $i, j \in \{1, \ldots, \mathrm{t}\}$, and I_t the $t \times t$ identity matrix.

Similar to SAFEOPT and other safe BO algorithms, we assume that the reward function lies in the RKHS of kernel k, i.e., $f \in H_k$. This assumption is, in general, non-restricting as many kernels satisfy the universal approximation property [13]. We can now obtain frequentist confidence intervals $Q_t(a)$ around the posterior mean μ_t that contain the ground truth f with high probability [11, Theorem 2]. We combine [11, Theorem 2] with data-dependent bounds from [14], as done in [15], to obtain

$$Q_t(a) \coloneqq \mu_t(a) \pm \left(B_t + \sqrt{2\sigma \log(1/\delta \det(1/\sigma K_t + I_t))}\right) \sigma_t(a), \tag{3}$$

with confidence parameter $\delta \in (0, 1)$. In (3), B_t is an over-estimation of the ground truth RKHS norm, i.e., $B_t \geq \|f\|_k$. The RKHS norm is given by $\|f\|_k^2 = \sum_{s=1}^{\infty} \sum_{t=1}^{\infty} \alpha_s \alpha_t k(x_s, x_t)$, where α are the coefficients and x are the center points of the RKHS function f. Notably, an under-estimation of the RKHS norm might lead to unsafe experiments (Figure 1), while a too conservative over-estimation might yield too cautious exploration and even premature stopping (Appendix A). In this paper, we compute a data-dependent B_t at each iteration t that over-estimates the RKHS norm $\|f\|_k$ with high probability. The data-driven RKHS norm over-estimation is the chief distinction between our approach and other safe BO algorithms like SAFEOPT [10] that *guess* the RKHS norm a priori.

Lipschitz constant. Besides knowing an upper bound on the RKHS norm, safe BO algorithms like SAFEOPT typically assume that an upper bound on the Lipschitz constant is known. We replace the Lipschitz constant with the RKHS norm over-estimation and the kernel (semi) metric

$$d_k(a, a') \coloneqq \sqrt{k(a, a) + k(a', a') - k(a, a') - k(a', a)}, \tag{4}$$

yielding an RKHS norm induced continuity [16, Proposition 3.1].

Safe exploration. Equivalent to SAFEOPT [10, 17], we define the contained set $C_t(a) \coloneqq C_{t-1}(a) \cap Q_t(a)$, $C_0 = \mathbb{R}$, lower bound $\ell_t(a) \coloneqq \min C_t(a)$, and upper bound $u_t(a) \coloneqq \max C_t(a)$ to quantify probabilistically whether a policy parameter a is safe. At each iteration t, we restrict function evaluations to a safe set $S_t \subseteq \mathcal{A}$ that only contains parameters a that are safe with high probability:

$$S_t \coloneqq \cup_{a \in S_{t-1}} \{a' \in \mathcal{A} | \ell_t(a) - B_t d_k(a, a') \geq h\}. \tag{5}$$

To start exploration, we assume that a set of initial safe samples $\emptyset \neq S_0 \subseteq \mathcal{A}$ is given and obtain a monotonically growing safe set S_t by sequentially augmenting the GP. Moreover, we define

$$M_t = \{a \in S_t | u_t(a) \geq \max_{a' \in S_t} \ell_t(a')\}, \tag{6}$$

$$G_t = \{a \in S_t | g_t(a) > 0\}, \quad g_t(a) \coloneqq \operatorname{card}(a' \in \mathcal{A} \setminus S_t | u_t(a) - B_t d_k(a, a') \geq h), \tag{7}$$

as the set of potential maximizers and potential expanders, respectively. At each iteration t, the next parameter a_{t+1} is given by the most uncertain parameter within $M_t \cup G_t$, i.e.,

$$a_{t+1} = \operatorname*{arg\,max}_{a \in M_t \cup G_t} \left(B_t + \sqrt{2\sigma \log(1/\delta \det(1/\sigma K_t + I_t))}\right) \sigma_t(a), \tag{8}$$

which results in safely balancing exploration and exploitation to solve (1).

3 Safe Bayesian optimization with RKHS norm over-estimation

Algorithm 1 summarizes the proposed safe BO algorithm with the RKHS norm over-estimation. In each iteration, we query the acquisition function and conduct an experiment with the newly acquired parameter. The acquisition function is described in Algorithm 2. First, we define the GP model given the current set of samples. Then, we compute an over-estimation of the RKHS norm by querying Algorithm 3, which we extensively explain in Section 3.1. Moreover, we compute the confidence intervals, the set of safe samples S_t, the set of potential maximizers M_t, and the set of potential expanders G_t. Finally, we return the most uncertain parameter within $M_t \cup G_t$ and its corresponding uncertainty. The acquisition function is reminiscent of SAFEOPT with the crux difference lying in the RKHS norm B_t (l. 2), where SAFEOPT *guesses* the RKHS norm a priori and maintains that guess. Hence, we naturally recover SAFEOPT by replacing the query of Algorithm 3 with an oracle.

In the remainder of this section, we present the RKHS norm over-estimation to compute B_t (Section 3.1), provide theoretical guarantees for $B_t \geq \|f\|_k$, derive novel confidence intervals for and prove safety of Algorithm 1 (Section 3.2), and extend Algorithm 1 by exploiting locality (Section 3.3).

Algorithm 1 Proposed safe BO algorithm with RKHS norm over-estimation

Require: k, $\mathcal{A}$, S_0, δ, κ, γ m, σ

Init: a_1 and y_1 samples corresponding to safe set S_0, $B_0 = \infty$
for $t = 1, 2, \ldots$ **do**
  $a_{t+1} \leftarrow$ Algorithm 2$(k, \mathcal{A}, S_{t-1}, \delta, \kappa, \gamma, m, \sigma, t)$ ▷ Acquisition function
  $y_{t+1} \leftarrow f(a_{t+1}) + \epsilon_t$ ▷ Conduct experiment
return Best safely evaluable parameter $a \in \mathcal{A}$

Algorithm 2 Acquisition function

Require: k, $\mathcal{A}$, S_{t-1}, δ, κ, γ, t, B_{t-1}, t, σ

Compute GP mean μ_t and covariance σ_t^2 given samples $a_{1:t}$ and $y_{1:t}$ ▷ (2)
$B_t \leftarrow$ Algorithm 3$(\gamma, \kappa, m, \mathcal{A}, k, B_{t-1}, a_{1:t}, y_{1:t}, k)$ ▷ RKHS norm over-estimation
Compute sets $Q_t(a)$, $C_t(a)$, and bounds $u_t(a), \ell_t(a)$ from samples $a_{1:t}, y_{1:t}$, and B_t ▷ (3)
if $t > 1$ **then** compute safe set S_t (5) **else** $S_t \leftarrow S_0$ ▷ (5)
Compute $\omega_t(a) \coloneqq \left(B_t + \sqrt{2\sigma \log(1/\delta \det(1/\sigma K_t + I_t))}\right) \sigma_t(a)$, and sets M_t, G_t ▷ (6), (7)
return $\arg\max_{a \in M_t \cup G_t} \omega_t(a)$, $\max_{a \in M_t \cup G_t} \omega_t(a)$ ▷ (8)

3.1 RKHS norm over-estimation

The RKHS norm over-estimation used in Algorithm 2 is based on two pillars: *(i)* a recurrent neural network (RNN) [18, 19] that predicts the RKHS norm for each iteration, and *(ii)* random RKHS functions that infer the potential behavior of the unknown reward function f.

RNN. We use an RNN to estimate the RKHS norm $\|f\|_k$ based on the current samples $a_{1:t}$ and $y_{1:t}$. Specifically, for each iteration, we compute the RKHS norm of the GP mean function $\|\mu_t\|_k$ and the reciprocal integral of the posterior covariance σ_t^2, which quantifies sampling density and store them as sequences. As the sampling density increases, the GP mean μ_t and its RKHS norm $\|\mu_k\|_k$ approximate the reward function f and its RKHS norm $\|f\|_k$ more closely. While the two sequences serve as the input to the RNN, we also require labels to train it. To this end, we optimize artificial RKHS functions $g \in H_k$, whose known RKHS norms $\|g\|_k$ serve as the labels for training the RNN, using our proposed safe BO algorithm. We provide more details on the RNN in Appendix B, including its architecture, the generation of training data, and its performance.

Random RKHS functions. The second pillar is the computation of random RKHS functions. In essence, the random RKHS functions $\rho_j \in H_k$, $j \in \{1, \ldots, \mathrm{m}\}$ capture the behavior of the unknown reward function f, as shown in Figure 2, which we exploit on top of the RNN to obtain theoretical guarantees on the RKHS norm over-estimation. Ideally, we would create random RKHS functions that capture the entire RKHS; however, this would require computing infinite sums. Hence, in implementation, we follow the pre-RKHS approach [20, Appendix C.1] to create random RKHS functions $\rho_j = \sum_{s=1}^{\hat{N}} \alpha_s k(\cdot, x_s)$, $\hat{N} \gg t$. Furthermore, we require the random RKHS functions to interpolate the given samples $y_{1:t}$ subject to σ-sub-Gaussian noise. Thus, the interpolating property determines the first $\alpha_{1:t}$ coefficients. Moreover, we assume that the first center points $x_{1:t}$ are equal to the parameters $a_{1:t}$. The remaining $\alpha_{t+1,\hat{N}}, x_{t+1,\hat{N}}$ are i.i.d. samples from uniform distributions with $x \in \mathcal{A}$ and $a \in [-\bar{\alpha}, \bar{\alpha}]$, introducing the required stochasticity. Subsequently, the random RKHS functions exhibit vastly different behavior for fewer samples and approach f for more samples (Figure 2), which will yield tighter RKHS norm over-estimations for an increasing sample density.

Algorithm. The RKHS norm over-estimation is summarized in Algorithm 3. First, we receive the RKHS norm estimation from the RNN, given the current set of samples. Second, we construct m i.i.d. random RKHS functions with known RKHS norms. Based on the return of the RNN and the RKHS norms of the random RKHS functions, we return B_t, which over-estimates $\|f\|_k$ with high probability. The explicit form of B_t becomes clear in Theorem 1.

Remark 1. *Our design choices decrease the space that is covered by the random RKHS functions. Nevertheless, the random RKHS functions in Figure 2 display a high degree of randomness, although they lie in a sub-space of the pre-RKHS from which f is generated, which supports the design choices.*

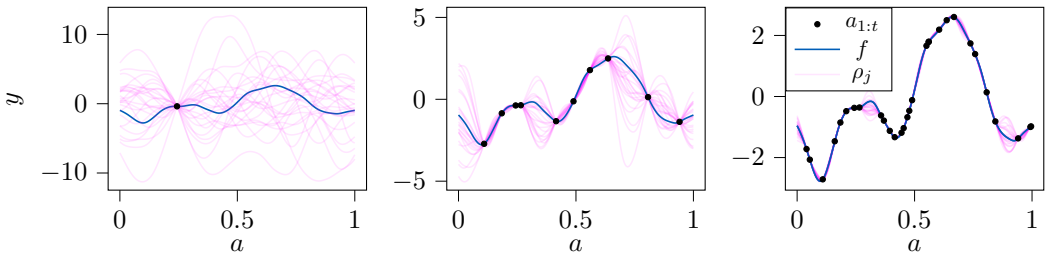

Figure 2: *Random RKHS functions.* The random RKHS functions approach the unknown reward function with more samples. We generated the plots with the Matérn32 kernel with length scale $\ell = 0.1$. The remaining hyperparameters were $\hat{N} = 500$, $\bar{\alpha} = 1$, and $\sigma = 10^{-2}$. The reward function f has 1000 random center points and coefficients, which were scaled to yield $\|f\|_k = 5$. We sampled the parameters $a_{1:t} \subseteq \mathcal{A}$ from a uniform distribution.

Algorithm 3 RKHS norm over-estimation

Require: γ, κ, m, $\mathcal{A}$, k, B_{t-1}, $a_{1:t}$, $y_{1:t}$, k

$B_t \leftarrow$ RKHS norm estimation given $a_{1:t}, y_{1:t}, k, \mathcal{A}$ with RNN

Construct m random RKHS functions $H_k \ni \rho_{t,j} \colon \mathcal{A} \to \mathbb{R}$, with $\|\rho_{t,j}\|_k$ given $a_{1:t}, y_{1:t}$

Sort random RKHS functions by ascending RKHS norm $\{\rho_{t,j}\}_{j=1}^{m}$

if $B_t < \|\rho_{t,m}\|_k$ **then**

$r \leftarrow \max_{r \in \{1,\dots,m-1\}} r$ subject to $\sum_{i=0}^{r} \binom{m}{i} \gamma^i (1-\gamma)^{m-i} \leq \kappa \ \wedge \ B_t < \|\rho_{t,m-r}\|_k$

if $B_t < B_{t-1}$ **then** $B_t \leftarrow B_{t-1}$

return B_t

Remark 2. *Although we integrate the RKHS norm over-estimation into* SAFEOPT*, it is equally applicable to any extension such as [17, 21–23]. Besides, the relevance of the RKHS norm goes beyond BO. It also appears in, e.g., statistics [24] or kernel-based function approximation [25].*

3.2 Theoretical analysis

In the following, we present theoretical guarantees on the RKHS norm over-estimation and Algorithm 1. First, we make an assumption on the inputs, the noise, and the kernel, akin to [11].

Assumption 1. *The kernel $k \colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}_{\geq 0}$ is symmetric, positive definite, and continuous. Moreover, the action sequence $\{a_t\}_{t=1}^{\infty}$ is an $\mathbb{R}^n$-valued discrete time stochastic process and a_t is $\mathcal{F}_{t-1}$-measurable $\forall t \geq 1$. The noise $\{\epsilon_t\}_{t=1}^{\infty}$ is a real-valued stochastic process and for some $\sigma \geq 0$ and all $t \geq 1$, ϵ_t is* (i) *$\mathcal{F}_t$-measurable and* (ii) *σ-sub-Gaussian conditionally on $\mathcal{F}_{t-1}$.*

Next, we introduce an assumption that connects the random RKHS functions and the reward function.

Assumption 2. *For any iteration $t \geq 1$, given $a_{1:t}, y_{1:t}$, the random RKHS functions $\rho_{t,j}, j \in \{1, \dots, \mathrm{m}\}$, and the reward function f are i.i.d. samples from the same probability space.*

Remark 3. *As $f \in H_k$, Assumption 2 is satisfied if $\rho_{t,j} \in H_k$. In practice, we have to restrict $\rho_{t,j}$ to a pre-RKHS $H_{0,k} \subseteq H_k$, as mentioned in Remark 1. However, the evaluation, as well as the numerical investigation in Section 5, in which we apply Algorithm 3 to 200 RKHS functions and quantify its performance, demonstrate the reliability of our proposed bounds.*

The following theorem is our main theoretical contribution and proves $B_t \geq \|f\|_k$ with high probability. Specifically, it shows that $B_t \geq \|f\|_k$ is probably approximately correct (PAC) [26].

Theorem 1 (RKHS norm over-estimation)**.** *Given Assumptions 1 and 2, for any iteration $t \geq 1$, $\gamma, \kappa \in (0,1)$, and $m \in \mathbb{N}$ such that $(1-\gamma)^{m-1}(1+\gamma(m-1)) \leq \kappa$, consider the B_t returned by Algorithm 3. With confidence at least $1-\kappa$, we have $B_t \geq \|f\|_k$ with probability at least $1-\gamma$.*

Proof. (Idea) First, we show that the RKHS norms of the ground truth and the random RKHS functions are i.i.d. random variables from the same probability space. Then, we formulate the RKHS norm over-estimation problem using a sampling-and-discarding scenario approach and obtain PAC bounds by leveraging [27, Theorem 2.1]. We provide a detailed proof in Appendix C.1. □

The following corollary lifts Theorem 3 to hold jointly for all iterations $t \geq 1$.

Corollary 1 (Lifting Theorem 1 to all iterations)**.** *Under the hypotheses of Theorem 1, receive B_t from Algorithm 3 at all iterations t. Then, with a confidence of at least $1-\kappa$, B_t over-estimates the ground truth RKHS norm $\|f\|_k$ jointly for all iterations $t \geq 1$ with a probability of at least $1-\gamma$.*

Proof. (Idea) First, we show that the discrete-time stochastic process $\{B_t\}_{t=1}^{T}, T \in \mathbb{N}$, containing the PAC RKHS norms is a supermartingale. Then, we use a standard stopping time criterion construction as in [28, Theorem 1]. We provide a detailed proof in Appendix C.2. □

Next, we present novel confidence intervals that contain the reward function f with high probability.

Theorem 2 (Confidence intervals)**.** *Under the same hypotheses as those of Corollary 1, let B_t be returned by Algorithm 3 $\forall t \geq 1$ with $\kappa, \gamma \in (0,1)$. Moreover, define $Q_t(a)$ as in* (3) *with any $\delta \in (0,1)$ and $C_t \coloneqq C_{t-1} \cap Q_t$ with $C_0 = \mathbb{R}$. Then, with confidence of at least $1-\kappa$, $f(a) \in C_t(a)$ holds jointly for all $a \in \mathcal{A}$ and for all $t \geq 1$ with probability of at least $(1-\gamma)(1-\delta)$.*

Proof. (Idea) First, we use the classic result from [11, Theorem 2] merged with bounds from [14]. Then, we combine this result with the PAC RKHS norm over-estimation from Corollary 1 by applying the law of total probability. We provide a detailed proof in Appendix C.3. □

Finally, we prove safety of the proposed safe BO algorithm with RKHS norm over-estimation.

Theorem 3 (Safety)**.** *Under the same hypotheses as those of Theorem 2, initialize Algorithm 1 with a safe set $S_0 \neq \emptyset$ such that $f(a) \geq h \; \forall a \in S_0$. Then, with confidence at least $1-\kappa$, $f(a_t) \geq h$ holds jointly for all $t \geq 1$ with a probability of at least $(1-\gamma)(1-\delta)$ when running Algorithm 1.*

Proof. (Idea) The proof is similar to the proof of [10, Theorem 1]. However, we replace the Lipschitz continuity from [10, Theorem 1] with an RKHS norm induced continuity from [16, Proposition 3.1] using the (semi) metric (4). Then, by the law of total probability, we combine the PAC RKHS norm over-estimation from Corollary 1 with the confidence intervals from Theorem 2 to show that all $a \in S_t$ are safe with high probability. We provide a detailed proof in Appendix C.4. □

3.3 Locality

Thus far, we proposed a safe BO algorithm with theoretical guarantees. At its heart lies the *data-driven computation of the RKHS norm*, which is required to, e.g., compute the safe set (5). The definition of the safe set implies that the algorithm explores in a neighborhood of already collected samples. Thus, we may not achieve the high sampling density on the entire parameter space that we would, following Figure 2, desire for a *tight* RKHS norm over-estimation. However, as we restrict exploration to the safe subset S_t of the parameter space $\mathcal{A}$, estimating the RKHS norm on $\mathcal{A} \setminus S_t$ is superfluous. Actually, it is precisely in unsafe areas where we expect non-smooth behavior and, hence, large RKHS norms. Thus, considering even the *true* global RKHS norm may yield overly conservative exploration, as also reported in [15]. Therefore,—inspired by local Lipschitz-based methods [29, 30]—we execute safe BO using sub-domains and localized RKHS norms while inheriting the theoretical guarantees derived for Algorithm 1.

Algorithm 4 summarizes the proposed localized safe BO algorithm with the data-driven RKHS norm over-estimation. We adopt an adaptive notion of locality by forming uniform local cubes around each sample $a \in a_{1:t}$. Specifically, we define N local cubes of width $(1, \ldots, \mathrm{N}) \cdot \Delta$ around each sample with hyperparameter $\Delta > 0$. Besides the local cubes, we preserve the global domain $\mathcal{A}$ and naturally recover Algorithm 1 by setting $N = 0$. We introduce the notation $\mathcal{C}_t \coloneqq \{0, \ldots, t \cdot N\}$ as the set of integers labeling the local cubes and the global domain and use the integer $c \in \mathcal{C}_t$ to refer to each object. At each iteration t and for each local cube $c \in \mathcal{C}_t$, we compute the local RKHS norm and determine a candidate parameter with (8). We choose the parameter for the next experiment as the most uncertain candidate parameter among all cubes.

Besides exploration benefits, the localized approach significantly improves the scalability of discretized BO algorithms like SAFEOPT. These discretized BO algorithms suffer from the curse of dimensionality since either the computational and memory complexities grow exponentially or we must accept a coarser discretization; the latter implying exponentially growing distances between

Algorithm 4 Proposed localized safe BO algorithm with RKHS norm over-estimation

Require: k, $\mathcal{A}$, S_0, δ, κ, γ m, σ, Δ, N

Init: a_1, y_1 samples corresponding to safe set S_0, $B_0 = \infty$
for $t = 1, 2, \dots$ **do**
  Compute $\mathcal{C}_t$ given t and N
  for $c \in \mathcal{C}_t$ **do** ▷ Iterate through sub-domains
    Determine $\mathcal{A}_c \subseteq \mathcal{A}$, $a_{1:t,c} \subseteq \mathcal{A}_c$, and $y_{1:t,c} \subseteq y_{1:t}$ given c and Δ
    $a_{t+1,c}$, $\omega_{t,c}(a_{t+1,c}) \leftarrow$ Algorithm 2$(k, \mathcal{A}_c, S_{t-1,c}, \delta, \kappa, \gamma, t, B_{t-1,c})$ ▷ Acquisition
  $a_{t+1} \leftarrow \arg\max_{a_{t+1,c}, c \in \mathcal{C}_t} \omega_{t,c}(a_{t+1,c})$ ▷ Most uncertain interesting parameter
  $y_{t+1} \leftarrow f(a_{t+1}) + \epsilon_t$ ▷ Conduct experiments
return Best safely evaluable parameter

the samples, in the worst case causing an empty safe set. The localized approach sequentially loops through each local cube when acquiring the next sample. This enables separate discretization in each local cube, which increases the discretization density and, therefore, simplifies exploration.

The following corollary formally states the inherited theoretical guarantees of Algorithm 4.

Corollary 2 (Localized safe BO)**.** *Choose any $N \in \mathbb{N}$, any $\Delta > 0$, consider any $t \geq 1$, and any $c \in \mathcal{C}_t$. Define the restriction f_c: $\mathcal{A}_c \subseteq \mathcal{A} \to \mathbb{R}$, $f_c(a) = f(a)$ for all $a \in \mathcal{A}_c$ and assume that $f_c \in H_k$, i.e., $\|f_c\|_k < \infty$. Moreover, let Assumption 2 hold for f_c. Then, the results from Theorem 1, Corollary 1, Theorem 2, and Theorem 3 are directly applicable for the local reward functions f_c and Algorithm 4, if the conditions therein are satisfied.*

Proof. Instead of deriving the mathematical statements only for the function f on the global domain $\mathcal{A}$, they are derived for f_c on $\mathcal{A}_c$ for all $c \in \mathcal{C}_t$ at any iteration $t \geq 1$. Since Algorithm 4 only samples from the corresponding safe sets, safety directly follows from Theorem 3. □

4 Related work

Next, we relate our safe BO algorithm with RKHS norm over-estimation to the state-of-the-art.

Safety. SAFEOPT [10] and its extensions [31, 21–23] require a tight upper bound on the RKHS norm of the unknown reward function to prove safety with high probability. The impracticability of this assumption has been addressed in [15] by proposing an algorithm similar to SAFEOPT, which instead relies on a priori upper bounds on *(i)* the noise and *(ii)* the Lipschitz constant of the unknown reward function; both of which are unknown and estimating the Lipschitz constant is similarly nontrivial as estimating RKHS norms. For GOOSE [32], another popular safe BO algorithm, [33] proposes a variant that approximates the Lipschitz constant, but without guarantees.

RKHS norm estimation. Only a few works tackle the RKHS norm estimation. References [34, 35] observe that the RKHS norm of the approximating function under-estimates the RKHS norm of the ground truth. Nevertheless, for safety guarantees, we require an over-estimation. Based on [34, 35], Reference [36] proposes a simple RKHS norm extrapolation, which empirically results in an upper bound, however, without any safety guarantees and in a noise-free setting with equidistant samples.

5 Experiments

In this section, we first provide a numerical investigation of the RKHS norm over-estimation. Then, we evaluate Algorithm 4 and compare it with SAFEOPT. Specifically, we illustrate the impact of estimating the RKHS norm instead of randomly guessing it in a one-dimensional toy experiment before comparing both algorithms on challenging RL benchmarks [37, 38]. Finally, we demonstrate the practicability of our algorithm by optimizing a controller for a real Furuta pendulum [39]. All experiments were conducted with hyperparameters $\sigma = 10^{-2}, \delta = 10^{-2}, \gamma = 10^{-1}, \kappa = 10^{-2}, \overline{\alpha} = 1$, $m = 1000$, and $\hat{N}_c = \max\{500\text{width}(\mathcal{A}_c), t+10\}$. Moreover, we shift and normalize the domains to yield $\mathcal{A} = [0, 1]^n$ and use the Matérn32 kernel with $\ell = 0.1$ unless stated otherwise. We provide videos for the RL and hardware experiments at `https://safeexploration.wordpress.com/`.

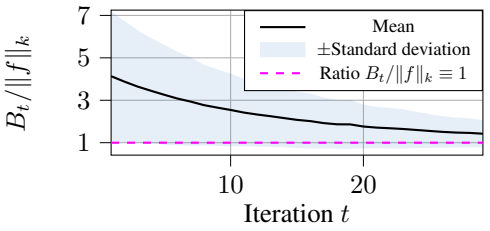

Figure 3: *Numerical investigation of the RKHS norm over-estimation.* For an increasing sample size, we receive tighter bounds.

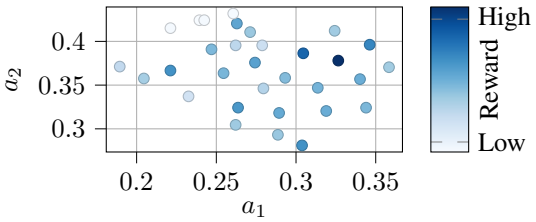

Figure 4: *Explored domain and rewards for the hardware experiment.* Algorithm 4 safely optimizes the controller for a Furuta pendulum.

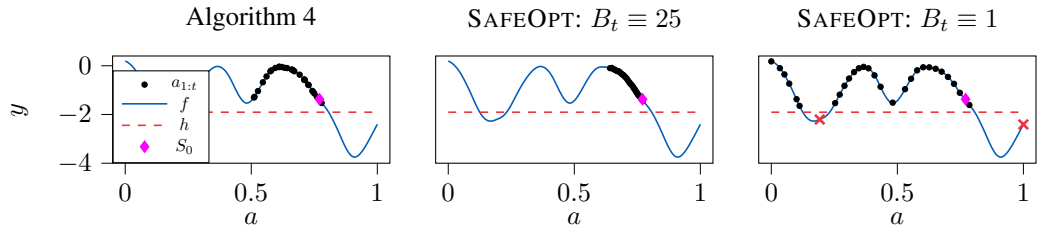

Figure 5: *Toy example to compare Algorithm 4 with* SafeOpt. Algorithm 4 (left) explores the domain and stays safe, while SafeOpt is either too conservative (center) or samples unsafely (right).

RKHS norm investigation. To test Corollary 1, we create 200 RKHS functions with RKHS norms sampled uniformly from $[1, 10]$. We sample the number of center points for each RKHS function uniformly from $[100, 1000]$, and scale the corresponding coefficients α to satisfy the predetermined $\|f\|_k$. At each iteration, we compute the over-estimations B_t using Algorithm 3 for each RKHS function f and append a new parameter sampled uniformly from $\mathcal{A}$. As already discussed in Section 3.1, we see in Figure 3 that the RKHS norm over-estimation gets tighter for an increasing sample set, supporting the sensibility of the proposed RKHS norm over-estimation. Crucially, in only two out of 200 cases did Algorithm 3 under-estimate the RKHS norm. As we chose $\gamma = 10^{-1}$ and $\kappa = 10^{-2}$, this is well within the guaranteed range.

Numerical experiments. To illustrate the benefits of our algorithm compared to SafeOpt, we let both maximize a synthetic function $f \in H_k$ generated with 1000 random center points x and coefficients α scaled to yield $\|f\|_k = 5$. For SafeOpt, we perform two runs, one with an over-estimation ($B_t \equiv 25$, center) and one with an under-estimation ($B_t \equiv 1$, right) of the RKHS norm. As shown in Figure 5, the former yields conservative exploration (crucially, it does not find the optimum within the given number of iterations), while the latter incurs failures (red crosses). In contrast, our algorithm (left) stays safe and finds the optimum. For Algorithm 4, we used $N = 5$ and $\Delta = 0.1$.

RL benchmarks. Next, we evaluate our algorithm and compare it to SafeOpt in challenging simulation benchmarks. In particular, we consider a sim-to-real setting, where no safety guarantees are required during simulation. Thus, we train policies in simulation using the soft actor-critic (SAC) algorithm [40] implemented in [41]. Those RL policies map from the states to the actions in $\mathbb{R}^n$ for the cart pole ($n = 1$), mountain car ($n = 1$), swimmer ($n = 2$), lunar lander ($n = 2$), and half cheetah ($n = 6$) environments [37, 38]. Then, to imitate real-world experiments, we manipulate the environments by, e.g., adding a wind disturbance for the lunar lander; see Appendix D for details. Thus, the policies learned with SAC still provide a safe starting point but are not optimal anymore. As we now must guarantee safety, we optimize these initial policies by learning an additive bias term $b \in \mathbb{R}^n$ using Algorithm 4 and SafeOpt. Figure 6 displays the reward development for the different environments. Algorithm 4 stays safe and learns a bias that improves the reward for all environments. For SafeOpt, a small RKHS norm leads to frequent safety violations (black crosses), which, e.g., correspond to the lunar lander crashing, whereas a large RKHS norm mostly yields conservative exploration and even premature stopping. Notably, even SafeOpt with a small

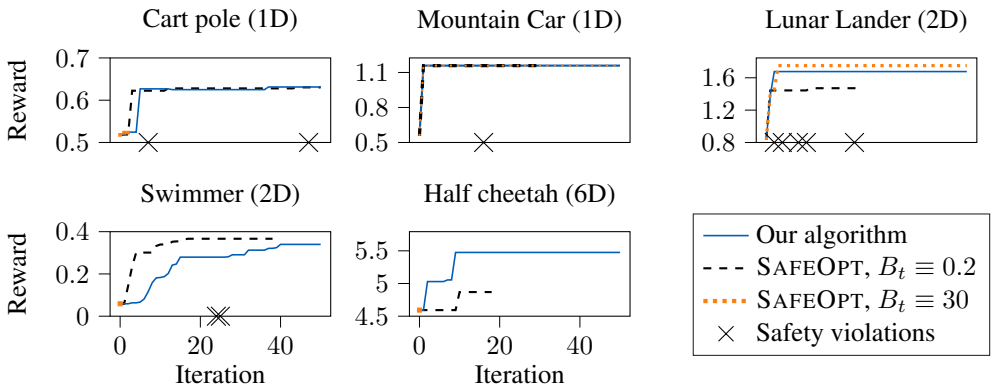

Figure 6: *RL benchmarks.* We optimize SAC policies by learning an additive bias in a sim-to-real inspired setting. Algorithm 4 exhibits better scalability, safety, and performance than SafeOpt. We plot the maximum reward encountered over iterations and mark violations of $h = 0$ with crosses.

RKHS norm fails to explore noticeably in the half cheetah environment, which is due to the coarse discretization in high dimensions, whereas our method improves scalability by exploiting locality and successfully improves the reward.

Hardware experiment. Lastly, we demonstrate the applicability of Algorithm 4 to real-world systems by optimizing the balancing controller of a Furuta pendulum [39]; see Appendix E for a visualization of the setup. We consider a similar experimental setup as [22], where the reward function corresponds to the control performance, and we tune the first two entries of a linear quadratic regulator (LQR). We execute Algorithm 4 with $\ell = 0.2$, $N = 3, \Delta = 0.15$ and we have $S_0 = [0.239, 0.424]^\top$ as an initial safe parametrization, see Figure 4. After 30 iterations, we significantly improved the controller performance while only conducting safe experiments, demonstrating that our algorithm is applicable to safety-critical real-world systems.

6 Conclusions

We presented a novel safe BO algorithm that learns an over-estimation of the RKHS norm from data including statistical guarantees. With that, it lifts the assumption of popular safe BO algorithms of knowing a tight upper bound on the RKHS norm a priori. We further developed novel confidence intervals for and proved safety of a safe BO algorithm with RKHS norm over-estimation. The proposed algorithm was also extended with an adaptive notion of locality and, thus, improved exploration and scalability. Finally, we demonstrated the benefits of our algorithm compared to SafeOpt in simulation and showed its practicability in a hardware experiment. Although we integrated the RKHS norm over-estimation and the locality into SafeOpt, both can equally be integrated into any modification or extension thereof. More importantly, we expect applications of the RKHS norm over-estimation to go beyond safe BO and open avenues for more realistic guarantees in general kernel-based methods or even for estimating Lipschitz constants with theoretical guarantees.

References

[1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[2] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[3] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971v6*, 2019.

[4] Roman Garnett. *Bayesian Optimization*. Cambridge University Press, 2023.

[5] Peter I Frazier. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[6] Carl Edward Rasmussen and Christopher K.I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[7] Rika Antonova, Akshara Rai, and Christopher G Atkeson. Deep kernels for optimizing locomotion controllers. *arXiv preprint arXiv:1707.09062*, 2017.

[8] Roberto Calandra, André Seyfarth, Jan Peters, and Marc Peter Deisenroth. Bayesian optimization for learning gaits under uncertainty. *Annals of Mathematics and Artificial Intelligence*, 76(1-2):5–23, 2016.

[9] Alonso Marco, Philipp Hennig, Jeannette Bohg, Stefan Schaal, and Sebastian Trimpe. Automatic LQR tuning based on Gaussian process global optimization. In *IEEE International Conference on Robotics and Automation*, pages 270–277, 2016.

[10] 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.

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

[12] 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.

[13] Charles A Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *Journal of Machine Learning Research*, 7(12), 2006.

[14] Yasin Abbasi-Yadkori. *Online learning for linearly parametrized control problems*. PhD thesis, 2013.

[15] Christian Fiedler, Johanna Menn, Lukas Kreisköther, and Sebastian Trimpe. On safety in safe Bayesian optimization. *arXiv preprint arXiv:2403.12948*, 2024.

[16] Christian Fiedler. Lipschitz and Hölder continuity in reproducing kernel Hilbert spaces. *arXiv preprint arXiv:2310.18078*, 2023.

[17] Felix Berkenkamp, Andreas Krause, and Angela P Schoellig. Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics. *Machine Learning*, 112(10):3713–3747, 2023.

[18] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[19] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[20] Christian Fiedler, Carsten W Scherer, and Sebastian Trimpe. Practical and rigorous uncertainty bounds for Gaussian process regression. In *AAAI Conference on Artificial Intelligence*, pages 7439–7447, 2021.

[21] 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, 2018.

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

[23] Bhavya Sukhija, Matteo Turchetta, David Lindner, Andreas Krause, Sebastian Trimpe, and Dominik Baumann. GoSafeOpt: Scalable safe exploration for global optimization of dynamical systems. *Artificial Intelligence*, 320:103922, 2023.

[24] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[25] Emilio Tanowe Maddalena, Paul Scharnhorst, and Colin N Jones. Deterministic error bounds for kernel-based learning techniques under bounded noise. *Automatica*, 134:109896, 2021.

[26] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[27] Marco C Campi and Simone Garatti. A sampling-and-discarding approach to chance-constrained optimization: feasibility and optimality. *Journal of Optimization Theory and Applications*, 148(2):257–280, 2011.

[28] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, 2011.

[29] Zhouxing Shi, Yihan Wang, Huan Zhang, J Zico Kolter, and Cho-Jui Hsieh. Efficiently computing local Lipschitz constants of neural networks via bound propagation. *Advances in Neural Information Processing Systems*, pages 2350–2364, 2022.

[30] Matt Jordan and Alexandros G Dimakis. Exactly computing the local Lipschitz constant of ReLU networks. *Advances in Neural Information Processing Systems*, pages 7344–7353, 2020.

[31] Felix Berkenkamp, Angela P Schoellig, and Andreas Krause. Safe controller optimization for quadrotors with Gaussian processes. In *IEEE International Conference on Robotics and Automation*, pages 491–496, 2016.

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

[33] Christopher König, Matteo Turchetta, John Lygeros, Alisa Rupenyan, and Andreas Krause. Safe and efficient model-free adaptive control via Bayesian optimization. In *IEEE International Conference on Robotics and Automation*, pages 9782–9788, 2021.

[34] Kazumune Hashimoto, Adnane Saoud, Masako Kishida, Toshimitsu Ushio, and Dimos V Dimarogonas. Learning-based symbolic abstractions for nonlinear control systems. *Automatica*, 146:110646, 2022. extended version on arxiv:1612.05327v3.

[35] Paul Scharnhorst, Emilio T. Maddalena, Yuning Jiang, and Colin N Jones. Robust uncertainty bounds in reproducing kernel Hilbert spaces: A convex optimization approach. *IEEE Transactions on Automatic Control*, 68(5):2848–2861, 2023.

[36] Abdullah Tokmak, Christian Fiedler, Melanie N. Zeilinger, Sebastian Trimpe, and Johannes Köhler. Automatic nonlinear MPC approximation with closed-loop guarantees. *arXiv preprint arXiv:2312.10199*, 2023.

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

[38] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012.

[39] Katsuhisa Furuta, M Yamakita, and S Kobayashi. Swing-up control of inverted pendulum using pseudo-state feedback. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 206(4):263–269, 1992.

[40] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870, 2018.

[41] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

[42] Giuseppe Carlo Calafiore and Marco C Campi. The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51(5):742–753, 2006.

[43] Rick Durrett. *Probability: Theory and Examples*. Cambridge University Press, 2019.

[44] Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer, 2008.

Appendix

Contents

A Additional figure for the introductory example

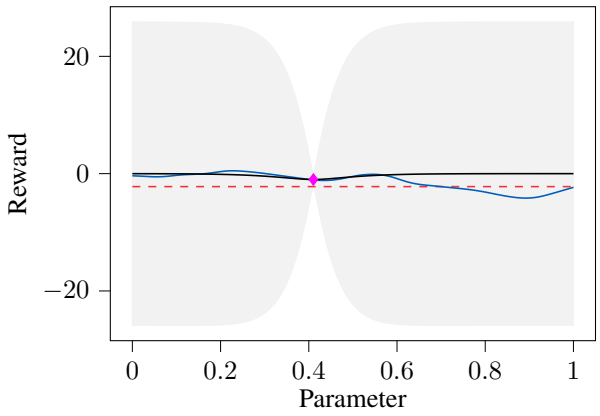

Figure 7: *Safe BO corresponding to Figure 1.* In this case, the RKHS norm is a conservative over-estimation. The safe BO algorithm cannot sample any parameter since none is safe with high probability. Hence, a conservative over-estimation of the RKHS norm is undesirable.

B Estimating RKHS norms with RNNs

We use a custom RNN to process data from two distinct input sequences: *(i)* from the RKHS norm of the GP mean μ_t; *(ii)* from the reciprocal integral of the GP posterior covariance σ_t^2. From these two sequences, the RNN extrapolates the unknown RKHS norm of the reward function $\|f\|_k$. For generating the training data and training the RNN, we used a cluster with 60 GB RAM and 20 cores.

Architecture. This model leverages two Long-Short-Term Memory RNN branches with twenty hidden layers, respectively. Moreover, each RNN branch contains two sigmoid and hyperbolic tangent activation functions, respectively. We use this custom RNN setup to capture temporal dependencies within each input stream independently before merging their representations to produce unified predictions.

Training data. Before training the RNN to estimate unknown RKHS norms $\|f\|_k$, we require training data. We generate training data by optimizing 10^3 artificial RKHS functions $g \in H_k$ using Algorithm 4. To generate g and by executing Algorithm 4, we use the Matérn32 kernel with lengthscale $\ell = 0.1$. We run Algorithm 4 with $\delta = 10^{-2}$, $\kappa = 10^{-2}$, $\gamma = 10^{-1}$, $\Delta = 10^{-1}$, and $N = 3$ for 50 iterations. To generate g, we first sample the number of center points uniformly from $[600, 1000]$ and sample the center points x uniformly from $\mathcal{A} = [0, 1]$. Then, we sample $\|g\|_k \in [0.5, 30]$ from a uniform distribution and scale the random coefficients α to satisfy the pre-determined $\|g\|_k$. When executing Algorithm 4, we generate training data from each local object $c \in \mathcal{C}_t$. Hence, we require the corresponding RKHS norm $\|g_c\|_k$ as the label, which is not directly inferred from the center points x and coefficients α of the function g. Thus, we densely discretize the function g_c for any $c \in C_t$ and any iteration t, and compute a heuristic RKHS norm $\|g_c\|_k$ using kernel interpolation; see e.g., [25] for the computation of the RKHS norm of the interpolating function.

Performance. The 10^3 functions g yielded $280 \cdot 10^3$ training samples for the RNN. We train the RNN with 100 epochs, a learning rate of 10^{-2}, and the ADAM optimizer, which took around $10\,\mathrm{min}$. We additionally preserved 20% of validation data. The root mean squared error on the validation data was approximately $5 \cdot 10^{-3}$.

C Proofs

C.1 Proof of Theorem 1

We prove the theorem by following a (sampling-and-discarding) scenario approach [27, 42]. To this end, we first need to show that under Assumption 2, the RKHS norms of the ground truth and all random RKHS functions are i.i.d. random variables conditioned on the samples.

Part I. Under Assumption 2, the stochasticity arises from the tail coefficients $\alpha_{t+1,\hat{N}}$ and the tail center points $x_{t+1:\hat{N}}$, which are i.i.d. uniform samples from $[-\bar{\alpha}, \bar{\alpha}]$ and $\mathcal{A}$, respectively. Choose any $t \geq 1$ with corresponding parameters $a_{1:t}$ and samples $y_{1:t}$ (and hence $x_{1:t}$ and $\alpha_{1:t}$). Consider the function

$$F(O) = \sqrt{\sum_{s=1}^{\hat{N}} \sum_{i=1}^{\hat{N}} \alpha_s \alpha_i k(x_s, x_i)}$$

that maps i.i.d. random center points and coefficients in $\mathbb{R}^{\hat{N}-t}$ to the resulting RKHS norm in $\mathbb{R}_{\geq 0}$ with corresponding Borel sets $\mathcal{R}^{\hat{N}-t}$ and $\mathcal{R}_{\geq 0}$, respectively, as σ-algebras. Hence, if F is measurable, then the RKHS norms are i.i.d. random variables [43, Theorem 1.3.5] from the same probability distribution; we prove measurability by contradiction.

Suppose that F is not measurable. Then, there exists an event $E \in \mathcal{R}_{\geq 0}$ with $E \ni F(O),\ O \in \mathbb{R}^{\hat{N}-t}$ such that $O \notin \mathcal{R}^{\hat{N}-t}$. This is a contradiction since $\mathcal{R}^{\hat{N}-t}$ is the Borel σ-algebra and, therefore, F is measurable. Hence, due to [43, Theorem 1.3.5], we deduce that $F(O)$ is a random variable and, thus, the RKHS norms are i.i.d. random variables from the same probability distribution.

Part II. Consider any iteration $t \geq 1$ and write the RKHS norm over-estimation as a constrained optimization problem

$$\begin{aligned} \min_{B_t^* \in \mathbb{R}_{\geq B_t}} \quad & B_t^* \\ \text{subject to} \quad & B_t^* \geq \|f\|_k. \end{aligned} \tag{9}$$

In this notation, B_t^* corresponds to the optimization variable and B_t to the value returned by the RNN. We could similarly consider the optimization domain $\mathbb{R}_{\geq 0}$. However, by lower-bounding B_t^* with the initial estimate obtained from the RNN, we introduce some conservatism. Clearly, Problem (9) is not solvable since $\|f\|_k$ is unknown. Hence, we formulate the optimization problem using the scenario approach [42] with m i.i.d. random RKHS functions $\rho_{t,j}$:

$$\begin{aligned} \min_{B_t^* \in \mathbb{R}_{\geq B_t}} \quad & B_t^* \\ \text{subject to} \quad & B_t^* \geq \|\rho_{t,j}\|_k \quad \forall j \in \{1, \ldots, m\}. \end{aligned} \tag{10}$$

We can use a scenario approach (10) to tackle Problem (9) since the RKHS norms are i.i.d. random variables from the same probability distribution [42]. Specifically, by solving (10), we obtain a solution that satisfies all m constraints, which, in return, yields a PAC solution for Problem (9). However, some of the random RKHS functions could be outliers with unreasonably high RKHS norms. To trade feasibility (constraint satisfaction with respect to all random RKHS functions) for performance (a smaller RKHS norm over-estimation), we follow a sampling-and-discarding scenario approach [27]. To this end, we formulate the following scalar optimization problem:

$$\begin{aligned} \min_{B_t^* \in \mathbb{R}_{\geq B_t}} \quad & B_t^* \\ \text{subject to} \quad & B_t^* \geq \|\rho_{t,j}\|_k \quad \forall i \in \{1, \ldots, m-r\} \\ & B_t^* < \|\rho_{t,j}\|_k \quad \forall j \in \{m-r+1, \ldots, m\}, \end{aligned} \tag{11}$$

i.e., the optimal solution violates r constraints corresponding to the r largest random RKHS norms.

We continue to map our problem to a sampling-and-discarding scenario approach, specifically to [27, Theorem 2.1]. Consider the probability space $(\mathbb{R}_{\geq 0}, \mathcal{R}_{\geq 0}, \mathbb{P})$. The probability space with m scenarios can be written as $(\mathbb{R}_{\geq 0}^m, \otimes_{j=1}^m \mathcal{R}_{\geq 0}, \mathbb{P}^m)$, equivalent to the setting in [27]. Before using [27, Theorem 2.1], we have to satisfy the following conditions:

(C1) The domain of the optimization problem is convex and closed.

(C2) The objective function is convex.

(C3) The feasible domain is convex and closed.

(C4) The optimization problem is feasible for $m < \infty$ with a feasibility domain with nonempty interior and unique solution.

(C5) The optimal solution violates all r discarded constraints almost surely.

We continue the proof in three different cases.

Case I, $B_t < \|\rho_{t,m}\|_k \wedge B_t \leq B_{t-1}$ In this case, the RKHS norm estimation returned by the RNN is smaller than the largest random RKHS norm and smaller than the previous PAC RKHS norm over-estimation. Condition (C1) is satisfied since $\mathbb{R}_{\geq B_t}$ is convex and closed for any $B_t \in \mathbb{R}$. Condition (C2) directly follows from having a linear objective function. Condition (C3) holds since the feasible domain is $[\|\rho_{t,m-r}\|_k, \|\rho_{t,m-r+1}\|_k) \subseteq \mathbb{R}_{\geq 0}$, with r computed in Algorithm 3 (l. 4). Moreover, Problem (11) is feasible for $m < \infty$ with a feasibility domain with nonempty interior and unique solution (C4). In fact, the solution of (11) is

$$B^\star_{t,m,r} = \max\{\|\rho_{t,m-r}\|_k, B_t\}, \tag{12}$$

explicitly denoting that the value depends on the number of scenarios m and the number of removed constraints $r < m$.

We now prove claim (C5), i.e., that $B^\star_{t,m,r}$ under-estimates the RKHS norms corresponding to $j = m - r + 1, \ldots, m$ in (11) almost surely. To this end, note that the RKHS norms are sorted in an ascending order and that $\|\rho_{t,j}\|_k \neq \|\rho_{t,i}\|_k, i, j \in \{1, \ldots, m\}, i \neq j$ almost surely. Since $B^\star_{t,m,r} = \max\{\|\rho_{t,m-r}\|_k, B_t\}$ with $B_t < \|\rho_{t,m-r}\|_k$ by Algorithm 3 (l. 4) and $\|\rho_{t,m-r}\|_k < \|\rho_{t,j}\|_k$, $\forall j \in \{m - r + 1, \ldots, m\}$ almost surely, the claim holds. Hence, we can use the result in [27, Theorem 2.1]:

$$\begin{aligned}&\mathbb{P}^m\left[(\|\rho_{t,1}\|_k, \ldots, \|\rho_{t,m}\|_k) \in \mathbb{R}^m_{\geq 0} : \mathbb{P}\left[\|f\|_k \in \mathbb{R}_{\geq 0}\colon B^\star_{t,m,r} \geq \|f\|_k\right] \geq 1 - \gamma\right] \\ &\geq 1 - \sum_{i=0}^{r} \binom{m}{i} \gamma^i (1-\gamma)^{m-i}.\end{aligned} \tag{13}$$

Inequality (13) provides PAC bounds on the constraint satisfaction for any unknown random variable in $\mathbb{R}_{\geq 0}$. Therefore, it probabilistically quantifies the constraint satisfaction of the optimal solution of (11) with respect to the unsolvable optimization problem (9), where we upper-bound the unknown RKHS norm $\|f\|_k$. Since Algorithm 3 requires

$$\sum_{i=0}^{r} \binom{m}{i} \gamma^i (1-\gamma)^{m-i} \leq \kappa$$

and sets $B_t = \max\{\|\rho_{t,m-r}\|_k, B_t\}$, we have

$$\mathbb{P}^m\left[(\|\rho_{t,1}\|_k, \ldots, \|\rho_{t,m}\|_k) \in \mathbb{R}^m_{\geq 0} : \mathbb{P}\left[\|f\|_k \in \mathbb{R}_{\geq 0}\colon B_t \geq \|f\|_k\right] \geq 1 - \gamma\right] \geq 1 - \kappa, \tag{14}$$

which concludes the proof for Case I.

Case II, $B_t \geq \|\rho_{t,m}\|_k \wedge B_t \leq B_{t-1}$ In this case, the RKHS norm estimation returned by the RNN is larger than the largest random RKHS norm and smaller than the previous PAC RKHS norm over-estimation. Then, we recover the classic scenario approach, i.e., we satisfy all m constraints, which can also be seen as a sampling-and-discarding scenario approach with $r = 0$ discarded constraints in Problem (11). Conditions (C1)-(C4) are satisfied equivalently to Case I, and Condition (C5) holds trivially since $r = 0$. The optimal solution of Problem (11) is given by

$$B^\star_{t,m,0} = B_t$$

and Algorithm 3 returns B_t as the PAC RKHS norm over-estimation.

Note that we choose γ, m, κ such that $(1-\gamma)^{m-1}(1 + \gamma(m-1)) \leq \kappa$ in Theorem 1. Since

$$\begin{aligned}\sum_{i=0}^{0} \binom{m}{i} \gamma^i (1-\gamma)^{m-i} &\leq \sum_{i=0}^{1} \binom{m}{i} \gamma^i (1-\gamma)^{m-i} \\ &= (1-\gamma)^{m-1}(1 + \gamma(m-1)) \\ &\leq \kappa,\end{aligned} \tag{15}$$

we can directly obtain PAC bounds for the optimal solution of the sampling-and-discarding scenario approach (11) with $r = 0$. Namely,

$$\begin{aligned}&\mathbb{P}^m\left[(\|\rho_{t,1}\|_k, \ldots, \|\rho_{t,m}\|_k) \in \mathbb{R}^m_{\geq 0} : \mathbb{P}\left[\|f\|_k \in \mathbb{R}_{\geq 0} \colon B_t \geq \|f\|_k\right] \geq 1-\gamma\right] \\ &\geq \sum_{i=0}^{0} \binom{m}{i} \gamma^i (1-\gamma)^{m-i} \overset{(15)}{\geq} 1-\kappa,\end{aligned}$$

which concludes the proof for Case II.

Case III, $B_t > B_{t-1}$ We now consider the case where the RKHS norm over-estimation at the previous iteration was tighter than the over-estimation at the current iteration. In this case, we choose

$$B_t = \min\{B_t, B_{t-1}\},$$

see Algorithm 3 (l. 5), with $B_0 = \infty$ by convention. The reason behind this choice is that if the estimation is PAC at iteration $t-1$, it is again PAC at iteration t. □

C.2 Proof of Corollary 1

Let $\{B_t\}_{t=1}^T,\ T \in \mathbb{N}$ be the discrete-time stochastic process containing the RKHS norm over-estimations for each iteration t. Since we choose $B_t = \min\{B_{t-1}, B_t\}$ in Algorithm 3, we have

$$B_t \leq B_{t-1} \leq \ldots \leq B_1 \quad \forall t \geq 1. \tag{16}$$

Moreover, let $\{\mathfrak{F}_t\}_{t=1}^T$ be a filtration with $\mathfrak{F}_t = \sigma(B_1, \ldots, B_t)$ the σ-algebras. Then, we have that $B_t \in \mathfrak{F}_t$ and due to (16), $\mathbb{E}[B_t] \leq B_1 < \infty$. Moreover,

$$\mathbb{E}[B_{t+1}|\mathfrak{F}_t] \leq B_t \leq B_1 \quad \forall t \geq 1,$$

follows from (16), i.e., $\{B_t\}_{t=1}^T$ is a supermartingale with respect to the filtration $\{\mathfrak{F}_t\}_{t=1}^T$ [43, Section 4.2]. Therefore, we can use a stopping-time construction for (super)martingales as done in [28, Theorem 1] and [11, Theorem 1].

Let us define the bad event

$$\mathfrak{B}_t = \{\omega \in \Omega \colon B_t < \|f\|_k\}$$

as under-estimating the ground truth RKHS norm $\|f\|_k$. Let τ' be the first time when the bad event $\mathfrak{B}_t$ happens, i.e.,

$$\tau'(\omega) \coloneqq \min\{t \geq 1 \colon \omega \in \mathfrak{B}_t\}$$

with $\min\{\emptyset\} = \infty$. Since

$$\bigcup_{t \geq 1} \mathfrak{B}_t = \{\omega \in \Omega \colon \tau'(\omega) < \infty\},$$

we have

$$\begin{aligned}\mathbb{P}\left[\cup_{t\geq 1}\mathfrak{B}_t\right] &= \mathbb{P}[\tau' < \infty] \\ &= \mathbb{P}\left[B_t < \|f\|_k, \tau' < \infty\right] \\ &\leq \mathbb{P}\left[B_t < \|f\|_k\right].\end{aligned} \tag{17}$$

In Theorem 1, we proved that $\mathbb{P}\left[B_t \geq \|f\|_k\right] \geq 1-\gamma$ holds with confidence $1-\kappa$ for any (fixed) $t \geq 1$. Therefore,

$$\mathbb{P}\left[B_t < \|f\|_k\right] \leq \gamma$$

holds with confidence $1-\kappa$ for any (fixed) $t \geq 1$, which with (17) implies that

$$\mathbb{P}\left[B_t \geq \|f\|_k\right] \geq 1-\gamma$$

holds with confidence $1-\kappa$ *jointly* for all $t \geq 1$. □

C.3 Proof of Theorem 2

First, we define the following events (the complementary event is denoted by the superscript $\perp$):

$\mathcal{C}_t$: It holds that $f(a) \in C_t(a)$ jointly for all $a \in \mathcal{A}$ and for all $t \geq 1$.
$\mathcal{Q}_t$: It holds that $f(a) \in Q_t(a)$ jointly for all $a \in \mathcal{A}$ and for all $t \geq 1$.
$\mathcal{B}_t$: It holds that $B_t \geq \|f\|_k$ jointly for $t \geq 1$.

The proof aims at providing a lower bound on the probability of occurrence of event $\mathcal{C}_t$. We start by investigating the probability of the event $\mathcal{Q}_t$ from which we can directly infer the probability of C_t. By the law of total probability, we can write

$$\begin{aligned}\mathbb{P}[\mathcal{Q}_t] &= \mathbb{P}[\mathcal{Q}_t|\mathcal{B}_t] \cdot \mathbb{P}[\mathcal{B}_t] + \mathbb{P}[\mathcal{Q}_t|\mathcal{B}_t^\perp] \cdot \mathbb{P}[\mathcal{B}_t^\perp] \\ &\geq \mathbb{P}[\mathcal{Q}_t|\mathcal{B}_t] \cdot \mathbb{P}[\mathcal{B}_t].\end{aligned}$$

By arguments of [11, Theorem 2], we have that

$$\mathbb{P}[\mathcal{Q}_t|\mathcal{B}_t] \geq 1 - \delta.$$

The stochasticity in [11, Theorem 2] arises from probabilistically upper-bounding the norm induced by a positive definite matrix of the noise vector $\epsilon_{1:t}$. The same case applies to our setting when conditioning on the event $\mathcal{B}_t$. However, different to [11, Theorem 2], we use the purely data-dependent upper bound from [14] to bound the noise, as done in [15]. The noise bound holds with a confidence of at least $1 - \delta$ for the noise in the present setting (Assumption 1). For details, we refer to the aforementioned works.

Moreover, with Corollary 1,

$$\mathbb{P}[\mathcal{B}_t] \geq 1 - \gamma$$

holds with a confidence of at least $1 - \kappa$. Therefore, with a confidence of at least $1 - \kappa$, it holds that

$$\mathbb{P}[\mathcal{Q}_t] \geq (1 - \gamma)(1 - \delta).$$

From [17, Corollary 7.1], we have

$$\mathbb{P}[\mathcal{C}_t] = \mathbb{P}[\mathcal{Q}_t],$$

and therefore

$$\mathbb{P}[\mathcal{C}_t] \geq (1 - \gamma)(1 - \delta),$$

with confidence $1 - \kappa$, which concludes the proof. □

C.4 Proof of Theorem 3

First, we present a Lipschitz-like continuity for RKHS functions for which we use the (semi) metric (4).

Lemma 1 (RKHS-induced continuity)**.** *[16, Proposition 3.1] Let all conditions in Theorem 3 hold and let B_t be returned by Algorithm 3. Then, jointly for any $a, a' \in \mathcal{A}$ and any $t \geq 1$, with a confidence least $1 - \kappa$,*

$$|h(a, i) - h(a', i)| \leq B_t d_k(a, a')$$

holds with probability of at least $1 - \gamma$.

Proof. With confidence $1 - \kappa$ and probability $1 - \gamma$,

$$\begin{aligned}|f(a) - f(a')| &= |\langle f, k(a, \cdot) - k(a', \cdot)\rangle_k| && \text{Reproducing property [44, Definition 4.18]} \\ &\leq \|f\|_k \sqrt{k(a, a) - k(a', a) - k(a, a') + k(a', a')} && \text{Cauchy-Schwarz inequality} \\ &\overset{(4)}{=} \|f\|_k d_k(x, x') \\ &\overset{\text{Cor.1}}{\leq} B_t d_k(x, x'),\end{aligned}$$

where $\langle f, g\rangle_k$ denotes the inner product between two functions in the RKHS of kernel k. Note that solely the last inequality introduces stochasticity and the previous steps hold deterministically. □

For each iteration $t \geq 1$, we are only allowed to sample within the safe set S_t (5). The following lemma exploits the definition of the safe set S_t to prove that we can guarantee safety with high probably for all iterations when only sampling within S_t.

Lemma 2. *Under the same hypotheses of Theorem 3, with confidence of at least* $1 - \kappa$*,*

$$\forall a \in S_t,\ f(a) \geq h$$

holds jointly for all iterations $t \geq 1$ *with probability of at least* $(1 - \delta)(1 - \gamma)$*.*

Proof. The lemma is akin to [10, Lemma 11]. However, we replace the assumption of knowing the true upper bound on the RKHS norm $\|f\|_k$ with the PAC RKHS norm over-estimation received by Algorithm 3. Furthermore, in contrast to [10], we do not require the Lipschitz constant and prove safety with high probability by exploiting the RKHS norm induced continuity formulated in Lemma 1 instead.

First, similar to the proof of Theorem 2, we introduce the following events (the complementary event is denoted by the superscript $\perp$):
Σ_t: It holds that $f(a) \geq h$ jointly $\forall a \in S_t, \forall t \geq 1$.
$\mathcal{C}_t$: It holds that $f(a) \in C_t(a)$ jointly for all $a \in \mathcal{A}$ and all $t \geq 1$.
$\mathcal{B}_t$: It holds that $B_t \geq \|f\|_k$ jointly for all $t \geq 1$.

Clearly, lower-bounding $\mathbb{P}[\Sigma_t]$ proves the lemma, which we formulate in three parts. In Part I, we compute $\mathbb{P}[\Sigma_t|\mathcal{B}_t, \mathcal{C}_t]$. In Part II, we lower-bound $\mathbb{P}[\mathcal{B}_t, \mathcal{C}_t]$ and finalize the proof by providing a lower bound on $\mathbb{P}[\Sigma_t]$ in Part III.

Part I: We prove that $\mathbb{P}[\Sigma_t|\mathcal{B}_t, \mathcal{C}_t] = 1$ by induction, equivalent to the proof of [10, Lemma 11].

Base case: In the first iteration, we set $S_1 \equiv S_0$, see Algorithm 2. Hence, by assumption, for all $a \in S_1, f(a) \geq h$ holds deterministically.
Induction step: Assume for some $t \geq 2$, $f(a) \geq h,\ \forall a \in S_{t-1}$. We show that $f(a) \geq h,\ \forall a \in S_{t-1}$ implies $f(a) \geq h,\ \forall a \in S_t$. Given the occurrence of both events $\mathcal{B}_t$ and $\mathcal{C}_t$, we have that $\forall a' \in S_t$, $\exists a \in S_{t-1}$ such that

$$\begin{aligned} h &\overset{(5)}{\leq} \ell_t(a) - B_t d_k(a, a') \\ &\overset{\text{Thm.2}}{\leq} f(a) - B_t d_k(a, a') \\ &\overset{\text{Lem.1}}{\leq} f(a'), \end{aligned}$$

i.e.,

$$\mathbb{P}[\Sigma_t|\mathcal{B}_t, \mathcal{C}_t] = 1. \tag{18}$$

Part II: By the formula for conditional probability, we have

$$\mathbb{P}[\mathcal{B}_t, \mathcal{C}_t] = \mathbb{P}[\mathcal{C}_t|\mathcal{B}_t] \cdot \mathbb{P}[\mathcal{B}_t].$$

From [11, Theorem 1], it follows that

$$\mathbb{P}[\mathcal{C}_t|\mathcal{B}_t] \geq 1 - \delta,$$

as argued in Theorem 2. Moreover,

$$\mathbb{P}[\mathcal{B}_t] \geq 1 - \gamma$$

with a confidence of at least $1 - \kappa$ follows from Corollary 1. Therefore,

$$\mathbb{P}[\mathcal{B}_t, \mathcal{C}_t] \geq (1 - \delta)(1 - \gamma) \tag{19}$$

with confidence of at least $1 - \kappa$.

Part III: By the law of total probability, we have that

$$\begin{aligned} \mathbb{P}[\Sigma_t] &= \mathbb{P}[\Sigma_t|\mathcal{C}_t, \mathcal{B}_t] \cdot \mathbb{P}[\mathcal{C}_t, \mathcal{B}_t] + \mathbb{P}[\Sigma_t|\mathcal{C}_t^\perp, \mathcal{B}_t] \cdot \mathbb{P}[\mathcal{C}_t^\perp, \mathcal{B}_t] \\ &\quad + \mathbb{P}[\Sigma_t|\mathcal{C}_t^\perp, \mathcal{B}_t^\perp] \cdot \mathbb{P}[\mathcal{C}_t^\perp, \mathcal{B}_t^\perp] + \mathbb{P}[\Sigma_t|\mathcal{C}_t, \mathcal{B}_t^\perp] \cdot \mathbb{P}[\mathcal{C}_t, \mathcal{B}_t^\perp] \\ &\geq \mathbb{P}[\Sigma_t|\mathcal{C}_t, \mathcal{B}_t] \cdot \mathbb{P}[\mathcal{C}_t, \mathcal{B}_t] \\ &\overset{(18)}{=} \mathbb{P}[\mathcal{C}_t, \mathcal{B}_t] \overset{(19)}{\geq} (1 - \delta)(1 - \gamma) \end{aligned}$$

holds with confidence at least $1 - \kappa$, which concludes the proof. □

Theorem 3 follows directly from Lemma 2. □

D Safe RL policy optimization in OpenAI Gym

In this section, we provide further details on the RL benchmark simulations. As discussed in Section 5, we trained the SAC algorithm [40] in various OpenAI Gym environments [37], in particular, the mountain car, the cart-pole system, the swimmer, the lunar lander, and the half-cheetah. We then alter specific physical properties within each environment to imitate real-world experiments, in which we utilize our proposed algorithm and SAFEOPT to optimize an action bias matching the dimensionality of the action space. We next state the remaining hyperparameters and detail how we alter the physical properties for the different environments. We conducted the experiments on a cluster with $60\,\mathrm{GB}$ RAM and 20 cores.

Mountain car (1D). We set $N = 3$, $\Delta = 10^{-1}$, and discretize the environment with 10^3 points. For the imitated real experiments, we reduce the power of the car from 0.015 to 0.013. The target is to reach the top of the mountain; any position before or after the goal point at the end of an episode was considered unsafe.

Cart-pole (1D). We set $N = 3$, $\Delta = 10^{-1}$, and discretize the environment with 10^3 points. For the imitated real experiments, we change the pole length from 0.6 to 0.8. The goal is to maintain the pole in an upright position; dropping the pole was considered unsafe.

Swimmer (2D). We set $N = 5$, $\Delta = 10^{-1}$, and discretize the environment with $5 \cdot 10^2$ points per dimension. For the imitated real experiments, we change the lengths of the "torso" and "back" links from 0.1 to 0.3. The goal is to achieve forward movement of the swimmer; any backward movement was considered unsafe.

Lunar lander (2D). We set $N = 5$, $\Delta = 10^{-1}$, and discretize the environment with $5 \cdot 10^2$ points per dimension. For the imitated real experiments, we add wind of velocity $3\,\mathrm{m\,s^{-1}}$. The goal was for the lander to descend and come to a complete rest; any instance of the lander tipping over or crashing was considered unsafe.

Half-cheetah (6D). We set $N = 10$, $\Delta = 5 \cdot 10^{-2}$, and discretize the environment with 8 points per dimension. For the imitated real experiments, we change the thickness of the back link from 0.046 to 0.066. The goal is to ensure forward movement without falling; any fall was considered unsafe.

E Hardware setup

We conducted the hardware experiment on an Ubuntu laptop with $32\,\mathrm{GB}$ RAM and an Intel Core i7-12700H processor. Figure 8 shows the setup of the Furuta pendulum.

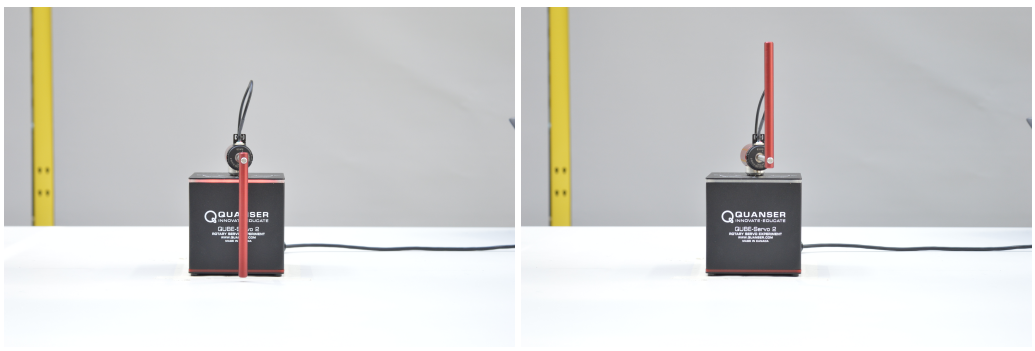

Figure 8: *Hardware setup*. The Furuta pendulum starts from a downward position (left) and is swung upright. Then, we use an LQR controller is used to balance the pendulum (right).