Safe exploration in reproducing kernel Hilbert spaces

Abdullah Tokmak
Department of Electrical
Engineering and Automation
Aalto University

Kiran G. Krishnan
Department of Electrical
Engineering and Automation
Aalto University

Thomas B. Schön
Department of
Information Technology
Uppsala University

Dominik Baumann
Department of Electrical
Engineering and Automation
Aalto University

Abstract

Popular safe Bayesian optimization (BO) algorithms learn control policies for safety-critical systems in unknown environments. However, most algorithms make a smoothness assumption, which is encoded by a known bounded 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, integrate the estimated RKHS norm into existing confidence intervals and show that we retain theoretical guarantees, 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 inverted pendulum, demonstrating improved performance, safety, and scalability compared to the state-of-the-art.

1 INTRODUCTION

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 (Lillicrap et al., 2019).

An alternative to RL for policy learning is combining Bayesian optimization (BO) (Garnett, 2023) with Gaussian process (GP) regression (Rasmussen and Williams, 2006). 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 drastically improves sample efficiency, as shown in numerous hardware experiments (Antonova et al., 2017; Calandra et al., 2016; Marco et al., 2016). 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 (Sui et al., 2015). SafeOpt, as well as other popular safe BO algorithms, assume that the reward function lies in a reproducing kernel Hilbert space (RKHS). Moreover, guaranteeing safety requires an additional smoothness assumption, which is encoded by knowing an 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, it is highly unrealistic since the RKHS is a potentially infinite-dimensional space, and it is unclear how to guess that upper bound for unknown reward functions. If we incorrectly specify the RKHS norm, i.e., if the true RKHS norm is larger than the bound we assume, safety guarantees may 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 this RKHS norm over-estimation into a safe BO algorithm reminiscent of SafeOpt, for which we prove safety with

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258. Copyright 2025 by the author(s).

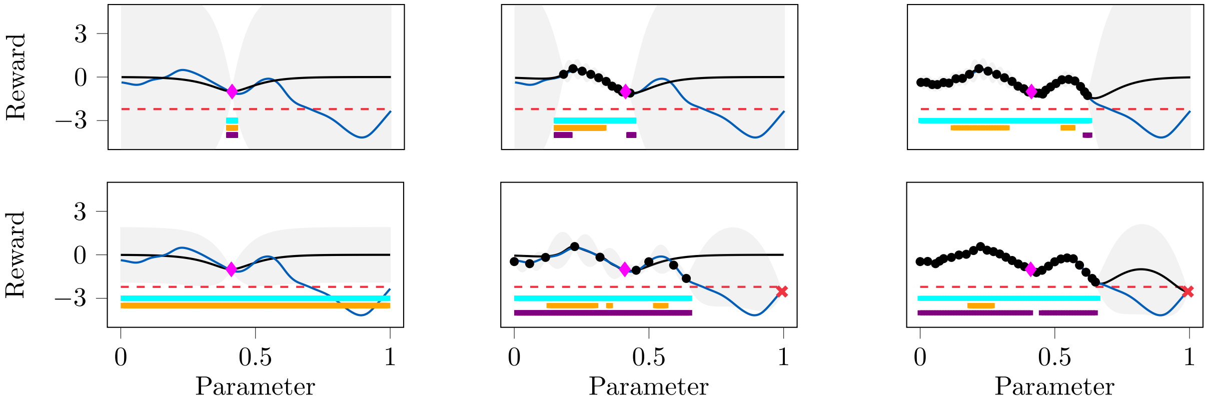

Figure 1: *Toy example of safe BO and the influence of the RKHS norm.* We aim to maximize the reward function (blue) while only sampling above the safety threshold (red dashed line). The predicted function (black line) is computed based on iteratively acquired samples (black dots, initial sample in magenta), and the gray shaded area shows the confidence intervals. 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).

high probability. Moreover, we extend our 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 significantly improve scalability by separately discretizing local sub-domains. We compare our algorithm with 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.[1]

2 PROBLEM SETTING AND PRELIMINARIES

We cast safe policy search as a constrained optimization problem, where the objective function quantifies the performance. We consider parameterized policies. The parameters, which could be controller parameters, 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 $a \in \mathcal{A}$ corresponding to reward values larger than a pre-defined safety threshold $h \in \mathbb{R}$. Thus, we write the optimization problem as

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

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 $a_{1:t} \coloneqq [a_1, \dots, a_t]^\top$ and the corresponding measurements by $y_{1:t}$. GP regression provides a natural tool to estimate f, as done in SafeOpt (Sui et al., 2015) and other BO algorithms (Srinivas et al., 2012). Given data $a_{1:t}$ and $y_{1:t}$ at each iteration t, the posterior GP mean and variance are

$$\mu_t(a) = k_t(a)^\top (K_t + \sigma^2 I_t)^{-1} y_{1:t}, \tag{2}$$
$$\sigma_t^2(a) = k(a,a) - k_t(a)^\top (K_t + \sigma^2 I_t)^{-1} k_t(a), \tag{3}$$

respectively (Rasmussen and Williams, 2006), where $k(a,a)$ is the kernel evaluated at $a \in \mathcal{A}$, $k_t(a) = [k(a,a_1), \dots, k(a,a_t)]^\top \in \mathbb{R}^t$ 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 $i,j \in \{1,\dots,\mathrm{t}\}$, 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 since many kernels satisfy the universal approximation property (Micchelli et al., 2006), i.e., even if $f \notin H_k$, there exists a $\tilde{f} \in H_k$ such that

[1] See https://github.com/tokmaka1/AISTATS_2025 for the code and https://safeexploration.wordpress.com/ for videos of the experiments.

$\sup_{a\in\mathcal{A}}|f(a) - \tilde{f}(a)| < \epsilon$ for all $\epsilon > 0$. Given this assumption, we can obtain frequentist confidence intervals $Q_t(a)$ around the posterior mean μ_t that contain f with high probability, i.e.,

$$\begin{aligned} Q_t(a) &\coloneqq \mu_t(a) \pm \beta_t\sigma_t(a), \\ \beta_t &= B_t + \sqrt{\sigma\log\det(I_t + K_t/\sigma) - 2\sigma\log(\delta)}, \end{aligned} \tag{4}$$

with confidence parameter $\delta \in (0, 1)$. We obtain (4) by combining Theorem 3.11 by Abbasi-Yadkori (2013) with Remark 3.13 by Abbasi-Yadkori (2013) as detailed in Appendix A. In (4), B_t is an over-estimation of the ground truth RKHS norm, i.e., $B_t \geq \|f\|_k = \sqrt{\sum_{s=1}^{\infty}\sum_{t=1}^{\infty}\alpha_s\alpha_t k(x_s, x_t)}$, where α are the coefficients and x are the center points of the RKHS function f (see Appendix B for the derivation). Notably, an under-estimation of the RKHS norm might lead to unsafe experiments (see Figure 1), while a too conservative over-estimation might yield too cautious exploration and even premature stopping (see Figure 8 in Appendix C). In this paper, we compute a data-dependent B_t at each iteration t that over-estimates $\|f\|_k$ with high probability. The data-driven RKHS norm over-estimation is the chief distinction between our approach and other safe BO algorithms like SafeOpt 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 a known Lipschitz constant. We replace the Lipschitz constant with an RKHS norm induced continuity using the kernel (semi)metric

$$d_k(a, a')^2 = k(a, a) + k(a', a') - k(a, a') - k(a', a), \tag{5}$$

which we derive in Lemma 1 in Appendix F.4.

Safe exploration Equivalent to SafeOpt, 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:

$$S_t \coloneqq \cup_{a\in S_{t-1}}\{a' \in \mathcal{A} | \ell_t(a) - B_t d_k(a, a') \geq h\}. \tag{6}$$

To start exploration, we assume that a set of initial safe samples $\emptyset \neq S_0 \subseteq \mathcal{A}$ is given. Moreover, we define

$$M_t \coloneqq \{a \in S_t | u_t(a) \geq \max_{a'\in S_t} \ell_t(a')\}, \tag{7}$$

$$\begin{aligned} G_t &\coloneqq \{a \in S_t | g_t(a) > 0\}, \\ g_t(a) &= \mathrm{card}(a' \in \mathcal{A} \setminus S_t | u_t(a) - B_t d_k(a, a') \geq h), \end{aligned} \tag{8}$$

as the set of potential safe maximizers and potential safe 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.,

$$a_{t+1} = \underset{a\in M_t\cup G_t}{\arg\max}\, \beta_t\sigma_t(a), \tag{9}$$

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

3 SAFE BO WITH RKHS NORM OVER-ESTIMATION

Algorithm 1 summarizes the proposed safe BO algorithm with the RKHS norm over-estimation. In each iteration, we determine the next sample with which we conduct a new experiment. The sample acquisition 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.

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

```
Require: $k$, $\mathcal{A}$, $S_0$, $\delta$, $\kappa$, $\gamma$ $m$, $\sigma$
1: Init: $a_1, y_1$ samples corresponding to $S_0$, $B_0 = \infty$
2: for $t = 1, 2, \ldots$ do
3:     $a_{t+1} \leftarrow$ Algorithm 2$(k, \mathcal{A}, S_{t-1}, \delta, \kappa, \gamma, m, \sigma, t)$
4:     $y_{t+1} \leftarrow f(a_{t+1}) + \epsilon_{t+1}$   ▷ Conduct experiment
5: return Best safely evaluable parameter $a \in \mathcal{A}$
```

Algorithm 2 Sample acquisition.

```
Require: $k$, $\mathcal{A}$, $S_{t-1}$, $\delta$, $\kappa$, $\gamma$, $t$, $B_{t-1}$, $t$, $\sigma$
1: Compute $\mu_t$ and $\sigma_t^2$ given $a_{1:t}, y_{1:t}$   ▷ (2), (3)
2: $B_t \leftarrow$ Algorithm 3$(\gamma, \kappa, m, \mathcal{A}, k, B_{t-1}, a_{1:t}, y_{1:t}, k)$
3: 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 $B_t$   ▷ (4)
4: if $t > 1$ then compute $S_t$ (6) else $S_t \leftarrow S_0$   ▷ (6)
5: Compute $\omega_t \coloneqq \beta_t\sigma_t$ and $M_t, G_t$   ▷ (7), (8)
6: return $\arg\max_{a\in M_t\cup G_t}\omega_t(a)$, $\max_{a\in M_t\cup G_t}\omega_t(a)$   ▷ (9)
```

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

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) (Hochreiter and Schmidhuber, 1997) that predicts the RKHS norm for each iteration, and *(ii)* random RKHS functions that simulate 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 variance 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 D, including its architecture, the generation of training data, its performance, and its role while executing the algorithm. We want to highlight that the RNN merely provides a *heuristic lower bound* on the estimated RKHS norm and solely acts as an additional layer of conservatism (see (12) in Appendix F.1); the hereafter introduced guarantees are *independent* from the estimation of the RNN. The RNN could be replaced by different function approximators; we choose an RNN to exploit the sequential nature of the inputs.

Random RKHS functions The second pillar is the computation of random RKHS functions with which we obtain theoretical guarantees on the RKHS norm over-estimation. In essence, the random RKHS functions $\rho_j \in H_k$, $j \in \{1, \dots, \mathrm{m}\}$ capture the behavior of the unknown reward function f, as shown in Figure 2. Ideally, we would create random RKHS functions that capture the entire RKHS; however, this would require computing infinite sums. Hence, in implementation, we follow a pre-RKHS approach as described in Appendix C.1 by Fiedler et al. (2021) to create random RKHS functions $\rho_j = \sum_{s=1}^{\hat{N}} \alpha_s k(\cdot, x_s)$, $\hat{N} \gg t$. 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 (see Figure 2), which will yield tighter RKHS norm over-estimations for an increasing sample density.

Algorithm 3 RKHS norm over-estimation

```
Require: $\gamma$, $\kappa$, $m$, $\mathcal{A}$, $k$, $B_{t-1}$, $a_{1:t}$, $y_{1:t}$, $k$
$B_t \leftarrow$ 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 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 $\sum_{i=0}^{r} \binom{m}{i} \gamma^i (1-\gamma)^{m-i} \leq \kappa \wedge B_t < \|\rho_{t,m-r}\|_k$
    $B_t \leftarrow \|\rho_{t,m-r}\|_k$
if $B_t < B_{t-1}$ then $B_t \leftarrow B_{t-1}$
return $B_t$
```

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. This supports the design choices. An alternative to the pre-RKHS approach is to work with orthonormal bases of RKHSs provided in Theorem 4.38 by Steinwart and Christmann (2008) for the squared-exponential kernel and by Tronarp and Karvonen (2024) for other translation-invariant kernels.*

Remark 2. *Although we integrate the RKHS norm over-estimation into* SafeOpt, *it applies equally to any extension such as by Berkenkamp et al. (2023); Sui et al. (2018); Sukhija et al. (2023). Besides, the relevance of the RKHS norm goes beyond BO. It appears in, e.g., statistics (Gretton et al., 2012) or kernel-based function approximation (Maddalena et al., 2021).*

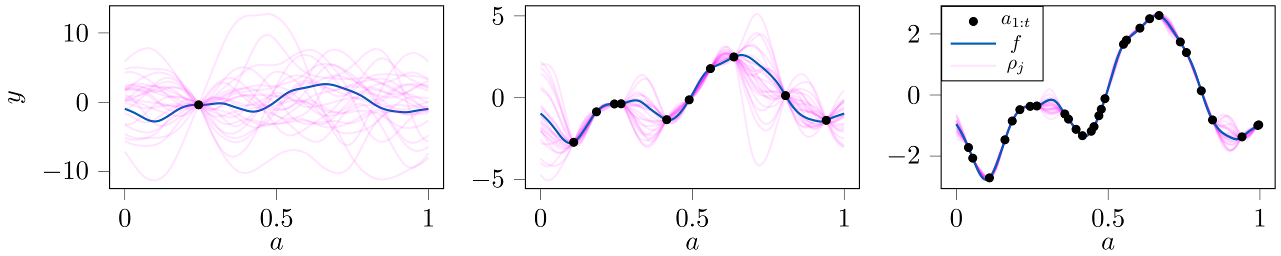

Figure 2: *Random RKHS functions.* The random RKHS functions approach the unknown reward function with more samples. We generated the plots with the Matérn32 kernel with $\ell = 0.1$. The remaining hyperparameters were $\hat{N} = 500$, $\bar{\alpha} = 1$, and $\sigma = 10^{-2}$. The reward function f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

3.2 Theoretical analysis

In the following, we present theoretical guarantees for the RKHS norm over-estimation and Algorithm 1. First, we make an assumption on the inputs, the noise, and the kernel, akin to Chowdhury and Gopalan (2017).

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 connect the RKHS norms of the random RKHS functions and the reward function.

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

We discuss Assumption 2 in Section 6 and in Appendix E. 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) (Shalev-Shwartz and Ben-David, 2014).

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 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) We formulate the RKHS norm over-estimation as a chance-constrained optimization problem, which we solve using a sampling-and-discarding scenario approach. We obtain PAC bounds by leveraging Theorem 2.1 by Campi and Garatti (2011). We provide a detailed proof in Appendix F.1. □

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

Corollary 1 (Lifting Theorem 1 to all iterations)**.** *Under the assumptions of Theorem 1, receive B_t from Algorithm 3 at all iterations t. Then, with confidence at least $1-\kappa$, B_t over-estimates the ground truth RKHS norm $\|f\|_k$ jointly for all iterations $t \geq 1$ with probability 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 Theorem 1 by Abbasi-Yadkori et al. (2011). We provide a detailed proof in Appendix F.2. □

Next, we integrate the RKHS norm over-estimation into existing confidence intervals that contain the reward function f with high probability.

Theorem 2 (Confidence intervals)**.** *Under the same assumptions 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 (4) with any $\delta \in (0,1)$ and $C_t \coloneqq C_{t-1} \cap Q_t$ with $C_0 = \mathbb{R}$. Then, with confidence 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 at least $(1-\gamma)(1-\delta)$.*

Proof. (Idea) We use classic confidence intervals by Abbasi-Yadkori (2013). Then, we combine this result with the PAC RKHS norm over-estimation from Corollary 1 by constructing a product probability space. We provide a detailed proof in Appendix F.3. □

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

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

Proof. (Idea) The proof is similar to the proof of Theorem 1 by Sui et al. (2015). However, we replace the Lipschitz continuity therein with an RKHS norm induced continuity from Proposition 3.1 by Fiedler (2023) using the (semi)metric (5). Then, we combine the confidence intervals from Theorem 2 with the definition of the safe set to prove that all $a \in S_t$ are safe with high probability. We provide a detailed proof in Appendix F.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 (6). 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.[2] Thus, considering even the *true* global RKHS norm may yield overly conservative exploration, as also reported by Fiedler et al. (2024). Therefore, we use local RKHS norms—inspired by local Lipschitz methods (Jordan and Dimakis, 2020)—to execute safe BO on sub-domains 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 $(1, \dots, \mathrm{N}) \cdot \Delta$ 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 $\mathcal{C}_t \coloneqq \{0, \dots, t \cdot N\}$ 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. Figure 3 illustrates the structure of the local cubes. At each iteration t and local cube $c \in \mathcal{C}_t$, we compute the local RKHS norm and determine a candidate parameter with (9). We choose the parameter for the next experiment as the most uncertain candidate parameter among all cubes.

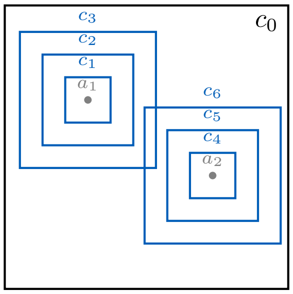

Figure 3: *Exemplary structure of the local cubes at iteration* $t = 2$. Each sample a_1, a_2 is the center of $N = 3$ local cubes of edge lengths $\Delta, 2\Delta$, and 3Δ, respectively. The global domain c_0 is depicted in black while the local cubes $c_1, \dots, c_6$ are illustrated in blue.

Algorithm 4 Localized safe BO algorithm with PAC RKHS norm over-estimation

```
Require: $k$, $\mathcal{A}$, $S_0$, $\delta$, $\kappa$, $\gamma$ $m$, $\sigma$, $\Delta$, $N$
Init: $a_1, y_1$ samples corresponding to $S_0$, $B_0 = \infty$
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 $\Delta$
        $a_{t+1,c}$, $\omega_{t,c}(a_{t+1,c}) \leftarrow$ Algorithm 2
    $a_{t+1} \leftarrow \arg\max_{a_{t+1,c}, c \in \mathcal{C}_t} \omega_{t,c}(a_{t+1,c})$
    $y_{t+1} \leftarrow f(a_{t+1}) + \epsilon_{t+1}$ ▷ Conduct experiments
return Best safely evaluable parameter
```

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 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* $f_c \colon \mathcal{A}_c \subseteq \mathcal{A} \to \mathbb{R}$, $f_c(a) = f(a)$ *for all* $a \in \mathcal{A}_c$ *and assume that* $f_c \in H_k$, *i.e.,* $\|f_c\|_k < \infty$. *Moreover, let all assumptions of Theorems 1-3 and Corollary 1 hold for* f_c. *Then, the results from Theorems 1-3 and Corollary 1 directly apply for the local reward functions* f_c *and Algorithm 4.*

[2] Let the reward be the distance to an equilibrium and consider the system $x_{k+1} = a x_k$. The system is stable at the equilibrium for, e.g., $a = 0.9999$ but moves away exponentially from the equilibrium for, e.g., $a = 1.0001$. Therefore, a small change in parameter a causes a significant change in the reward and thus a large local RKHS norm.

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.

SafeOpt (Sui et al., 2015) and its extensions (Berkenkamp et al., 2016; Sui et al., 2018) require an 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 by Fiedler et al. (2024) 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. Berkenkamp et al. (2019) investigate BO with unknown hyperparameters by decreasing the length scale and increasing the RKHS norm to construct an iteratively richer RKHS that will eventually contain the ground truth. However, they do not provide safety nor guarantee *when* the RKHS contains the function.

Only a few works tackle the RKHS norm estimation. Hashimoto et al. (2022) and Scharnhorst et al. (2023) 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 these works, Tokmak et al. (2023) propose a simple RKHS norm extrapolation, which empirically results in an upper bound, however, without statistical guarantees and in a noise-free setting with equidistant samples. Moreover, Karvonen (2022) empirically estimates a computable error bound by using the RKHS norm of the GP mean, similar to the idea presented in Equation (10) by Fasshauer (2011). Both works do not provide any guarantees.

The only other work we are aware of that develops a safe BO algorithm capable of estimating the RKHS norm with theoretical guarantees is Tokmak et al. (2024). In contrast, we prove safety guarantees for the resulting safe BO algorithm (Theorem 3) instead of only providing statistical guarantees for the RKHS norm over-estimation. Moreover, we obtain a tighter RKHS norm over-estimation by using a sampling-and-discarding scenario approach instead of Hoeffding's inequality. We compare the tightness in Appendix G. Further, our algorithm is *significantly* more scalable by using an *adaptive* notion of locality, allowing for separate discretization in each sub-domain. We elaborate on the improved scalability through adaptive locality in Appendix H. Finally, we work under less restricting and more interpretable assumptions as discussed in Appendix E.

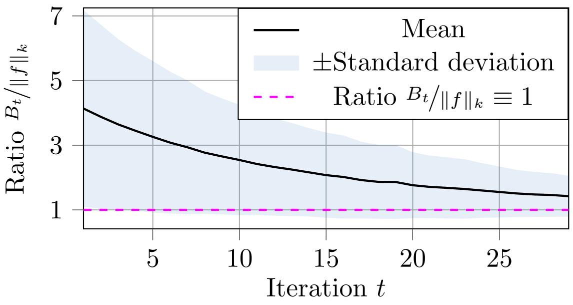

Figure 4: *Numerical investigation of the RKHS norm over-estimation.* For an increasing sample size, we receive tighter bounds.

5 EXPERIMENTS

In this section, we first numerically investigate the scenario approach. 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. Finally, we demonstrate the practicability of our algorithm by optimizing a controller for a real Furuta pendulum (Furuta et al., 1992). 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.

Scenario approach 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 pre-determined $\|f\|_k$. 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 $\mathcal{A}$. We present the numerical investigation in Figure 4. As already discussed in Section 3.1, we see 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 specified in Corollary 1. Moreover, we investigate the required computation time for the scenario approach in Appendix I and conduct an ablation study in Appendix J.

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 $\|f\|_k = 5$, which we present in Figure 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. 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 $\Delta = 0.1$. We provide ablation studies with different kernels and different locality parameters in Appendices K and L, respectively.

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 (Haarnoja et al., 2018; Raffin et al., 2021). 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 ($n = 2$), half cheetah ($n = 6$), and ant ($n = 8$) environments (Brockman et al., 2016; Todorov et al., 2012). Then, to imitate real-world experiments, we manipulate the environments by, e.g., adding a wind disturbance for the lunar lander; see Appendix M 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 shows the rewards over iterations 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 or premature stopping. Importantly, even SafeOpt with a small RKHS norm fails to explore noticeably in the half cheetah and ant environments, 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 Furuta et al. (1992); see Appendix N for a visualization of the setup. We consider a similar experimental setup as Baumann et al. (2021), where the reward function corresponds to the control performance, and we tune the first two entries of a state-feedback controller. We execute Algorithm 4 with $\ell = 0.2$, $N = 3, \Delta = 0.15$ and we have $S_0 = [0.239, 0.424]^\top$. After 30 iterations, we explored the domain to significantly improve the controller performance while only conducting safe experiments, as shown in the video and in Figure 7. This demonstrates that our algorithm is applicable to safety-critical real-world systems.

6 LIMITATIONS

In this section, we discuss the limitations of our contributions, specifically Assumption 2. Assumption 2 essentially states that f and $\rho_{t,j}$ are i.i.d. samples from the same—potentially unknown—probability space. However, in the frequentist setting, f is a fixed sample generated by *nature's probability space*. Hence, even if we would be able to sample from the entire RKHS H_k (see Remark 1), in practice, we *never* have access to nature's probability space and, thus, Assumption 2 implies a sampling oracle. Hence, by generating random RKHS functions, we approximate that probability space and impose a *prior* on f. Therefore, we essentially are in the Bayesian setting. However, mixing both frequentist and Bayesian methods is fairly common (Bayarri and Berger, 2004; Baggio et al., 2022). Moreover, assuming an a priori tight upper bound on $\|f\|_k$ (Sui et al., 2015) or assuming that the expected value of the RKHS norms of the random RKHS functions over-estimates $\|f\|_k$ (Tokmak et al., 2024) restricts nature's function space and also imposes prior knowledge on f. Also, we explain the mathematical meaning of Assumption 2 in Appendix E and contrast it further to the assumptions made by Sui et al. (2015) and Tokmak et al. (2024). In conclusion, we remove the *a priori guess* on the RKHS norm by introducing Assumption 2, enabling us to incorporate data into the RKHS norm bound. Hence, we can cover a *rich set of functions* and adjust the bounds as we gather more data (see Figures 2 and 4), yielding reliable bounds in practice with random RKHS functions from sup- or sub-RKHSs of the RKHS of the ground truth.

7 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 fur-

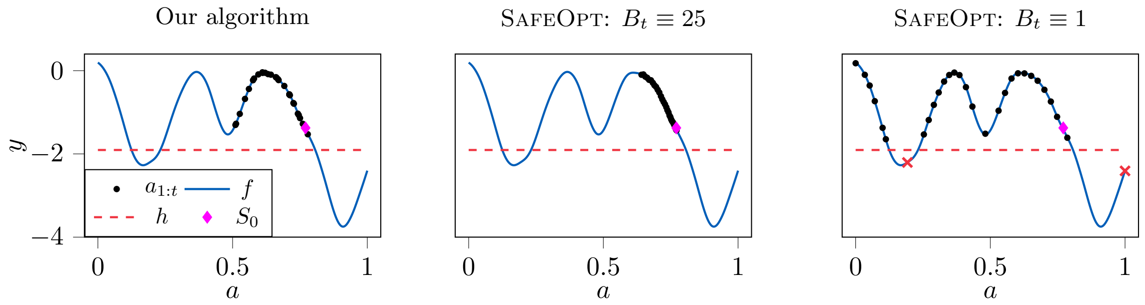

Figure 5: *Toy example to compare Algorithm 4 with* SafeOpt. Algorithm 4 (left) explores the domain and stays safe, while SafeOpt is either too conservative (center) or samples unsafely (right).

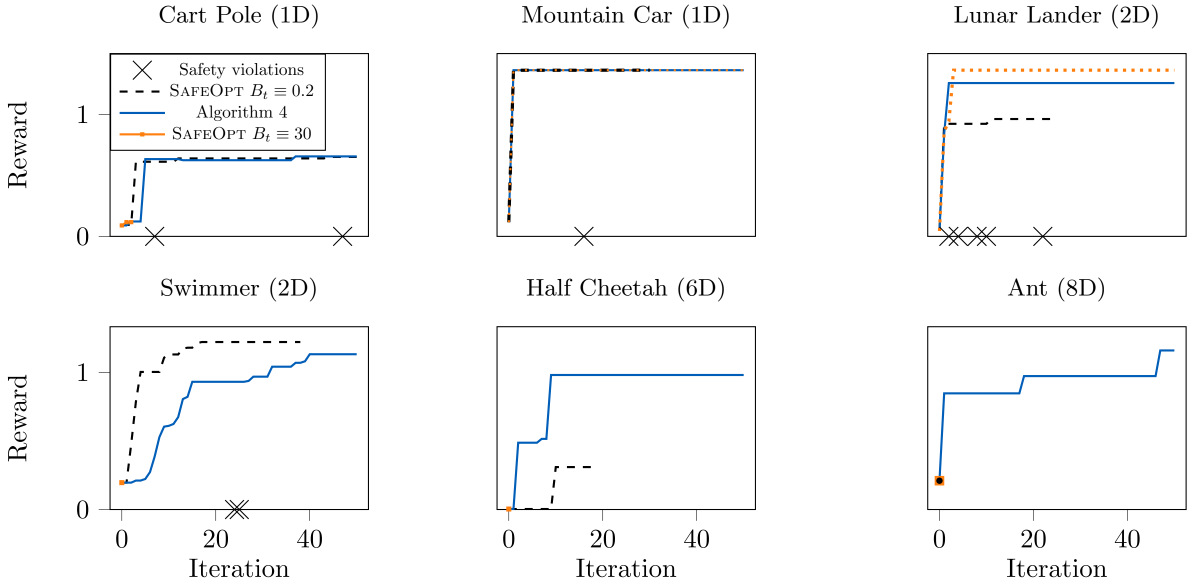

Figure 6: *RL benchmarks.* We optimize SAC policies by learning an additive bias in a sim-to-real inspired setting. Algorithm 4 exhibits better scalability, safety, and performance than SafeOpt. We plot the maximum scaled reward encountered over iterations and mark violations of $h = 0$ with crosses.

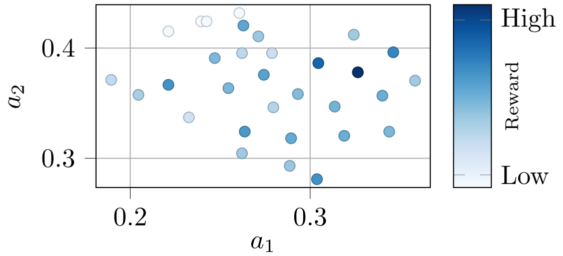

Figure 7: *Explored domain and rewards for the hardware experiment.* Algorithm 4 safely optimizes the controller for a Furuta pendulum.

ther proved safety of the developed safe BO algorithm with RKHS norm over-estimation. The proposed algorithm was extended with an adaptive notion of locality and, thus, improved exploration and scalability. We demonstrated the benefits of our algorithm compared to SafeOpt in simulation and showed that it can successfully handle real-world experiments. 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 for estimating e.g., Lipschitz constants with theoretical guarantees. Future work includes proving optimality of Algorithm 4, investigating regret bounds, and disentangling the constraints from the reward function.

Acknowledgments

We thank Harsha V. Guda for technical support. We also wish to acknowledge CSC – IT Center for Science, Finland, for computational resources. This research was partially supported by *Kjell och Märta Beijer Foundation*.

References

Abbasi-Yadkori, Y. (2013). *Online learning for linearly parametrized control problems*. PhD thesis.

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*.

Antonova, R., Rai, A., and Atkeson, C. G. (2017). Deep kernels for optimizing locomotion controllers. *arXiv preprint arXiv:1707.09062*.

Baggio, G., Carè, A., Scampicchio, A., and Pillonetto, G. (2022). Bayesian frequentist bounds for machine learning and system identification. *Automatica*, 146:110599.

Baumann, D., Kowalczyk, K., Tiels, K., and Wachel, P. (2023). A computationally lightweight safe learning algorithm. In *IEEE Conference on Decision and Control*, pages 1022–1027.

Baumann, D., Marco, A., Turchetta, M., and Trimpe, S. (2021). GoSafe: Globally optimal safe robot learning. In *IEEE International Conference on Robotics and Automation*, pages 4452–4458.

Bayarri, M. J. and Berger, J. O. (2004). The interplay of bayesian and frequentist analysis. *Statistical Science*.

Berkenkamp, F., Krause, A., and Schoellig, A. P. (2023). Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics. *Machine Learning*, 112(10):3713–3747.

Berkenkamp, F., Schoellig, A. P., and Krause, A. (2016). Safe controller optimization for quadrotors with Gaussian processes. In *IEEE International Conference on Robotics and Automation*, pages 491–496.

Berkenkamp, F., Schoellig, A. P., and Krause, A. (2019). No-regret bayesian optimization with unknown hyperparameters. *Journal of Machine Learning Research*, 20(50):1–24.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). OpenAI Gym. *arXiv preprint arXiv:1606.01540*.

Calafiore, G. C. and Campi, M. C. (2006). The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51(5):742–753.

Calandra, R., Seyfarth, A., Peters, J., and Deisenroth, M. P. (2016). Bayesian optimization for learning gaits under uncertainty. *Annals of Mathematics and Artificial Intelligence*, 76(1-2):5–23.

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

Chowdhury, S. R. and Gopalan, A. (2017). On kernelized multi-armed bandits. In *International Conference on Machine Learning*, pages 844–853.

Durrett, R. (2019). *Probability: Theory and Examples*. Cambridge University Press.

Fasshauer, G. E. (2011). Positive definite kernels: past, present and future. *Dolomites Research Notes on Approximation*, 4:21–63.

Fiedler, C. (2023). Lipschitz and Hölder continuity in reproducing kernel Hilbert spaces. *arXiv preprint arXiv:2310.18078*.

Fiedler, C., Menn, J., Kreisköther, L., and Trimpe, S. (2024). On safety in safe Bayesian optimization. *arXiv preprint arXiv:2403.12948*.

Fiedler, C., Scherer, C. W., and Trimpe, S. (2021). Practical and rigorous uncertainty bounds for Gaussian process regression. In *AAAI Conference on Artificial Intelligence*, pages 7439–7447.

Furuta, K., Yamakita, M., and Kobayashi, S. (1992). 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.

Garnett, R. (2023). *Bayesian Optimization*. Cambridge University Press.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012). A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870.

Hashimoto, K., Saoud, A., Kishida, M., Ushio, T., and Dimarogonas, D. V. (2022). Learning-based symbolic abstractions for nonlinear control systems. *Automatica*, 146:110646. extended version on arxiv:1612.05327v3.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Jordan, M. and Dimakis, A. G. (2020). Exactly computing the local Lipschitz constant of ReLU networks. *Advances in Neural Information Processing Systems*, pages 7344–7353.

Karvonen, T. (2022). Error bounds and the asymptotic setting in kernel-based approximation. *Dolomites Research Notes on Approximation*, 15(3):65–77.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2019). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971v6*.

Maddalena, E. T., Scharnhorst, P., and Jones, C. N. (2021). Deterministic error bounds for kernel-based learning techniques under bounded noise. *Automatica*, 134:109896.

Marco, A., Hennig, P., Bohg, J., Schaal, S., and Trimpe, S. (2016). Automatic LQR tuning based on Gaussian process global optimization. In *IEEE International Conference on Robotics and Automation*, pages 270–277.

Micchelli, C. A., Xu, Y., and Zhang, H. (2006). Universal kernels. *Journal of Machine Learning Research*, 7(12).

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., and Dormann, N. (2021). Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8.

Rasmussen, C. E. and Williams, C. K. (2006). *Gaussian Processes for Machine Learning.* MIT Press.

Romao, L., Papachristodoulou, A., and Margellos, K. (2022). On the exact feasibility of convex scenario programs with discarded constraints. *IEEE Transactions on Automatic Control*, 68(4):1986–2001.

Scharnhorst, P., Maddalena, E. T., Jiang, Y., and Jones, C. N. (2023). Robust uncertainty bounds in reproducing kernel Hilbert spaces: A convex optimization approach. *IEEE Transactions on Automatic Control*, 68(5):2848–2861.

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms.* Cambridge University Press.

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

Steinwart, I. and Christmann, A. (2008). *Support Vector Machines.* Springer.

Sui, Y., Gotovos, A., Burdick, J., and Krause, A. (2015). Safe exploration for optimization with Gaussian processes. In *International Conference on Machine Learning*, pages 997–1005.

Sui, Y., Zhuang, V., Burdick, J., and Yue, Y. (2018). Stagewise safe Bayesian optimization with Gaussian processes. In *International Conference on Machine Learning*, pages 4781–4789.

Sukhija, B., Turchetta, M., Lindner, D., Krause, A., Trimpe, S., and Baumann, D. (2023). GoSafeOpt: Scalable safe exploration for global optimization of dynamical systems. *Artificial Intelligence*, 320:103922.

Todorov, E., Erez, T., and Tassa, Y. (2012). Mujoco: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033.

Tokmak, A., Fiedler, C., Zeilinger, M. N., Trimpe, S., and Köhler, J. (2023). Automatic nonlinear MPC approximation with closed-loop guarantees. *arXiv preprint arXiv:2312.10199*.

Tokmak, A., Schön, T. B., and Baumann, D. (2024). PACSBO: Probably approximately correct safe Bayesian optimization. *arXiv preprint arXiv:2409.01163*.

Tronarp, F. and Karvonen, T. (2024). Orthonormal expansions for translation-invariant kernels. *Journal of Approximation Theory*, page 106055.

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

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

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

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

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

Safe exploration in reproducing kernel Hilbert spaces: Appendix

Contents

A DERIVATION OF THE CONFIDENCE INTERVALS (4)

In this section, we derive the confidence intervals that were initially presented in the dissertation by Abbasi-Yadkori (2013). These data-dependent bounds have gained increasing interest, see e.g., Fiedler et al. (2021) or Fiedler et al. (2024).

Writing Theorem 3.11 and Remark 3.13 by Abbasi-Yadkori (2013) using our notation yields

$$|f(\cdot) - \mu_t(\cdot)| \leq \|\mathfrak{m}\|_{\overline{V}_t^{-1}} \left(\sigma \sqrt{2 \log \left(\frac{\det(I_t + K_t \sigma)^{\frac{1}{2}}}{\delta} \right)} + \sqrt{\sigma} B_t \right),$$

with

$$\|\mathfrak{m}\|_{\overline{V}_t^{-1}} = \frac{1}{\sqrt{\sigma}} \sigma_t(\cdot),$$

which gives

$$\begin{aligned}
|f(\cdot) - \mu_t(\cdot)| &\leq \frac{1}{\sqrt{\sigma}} \left(\sigma \sqrt{2 \log \left(\frac{\det(I_t + K_t/\sigma)^{\frac{1}{2}}}{\delta} \right)} + \sqrt{\sigma} B_t \right) \sigma_t(\cdot) \\
&= \left(\sqrt{\sigma} \sqrt{2 \log \left(\frac{\det(I_t + K_t/\sigma)^{\frac{1}{2}}}{\delta} \right)} + B_t \right) \sigma_t(\cdot) \\
&= \left(\sqrt{2\sigma \log \left(\frac{\det(I_t + K_t/\sigma)^{\frac{1}{2}}}{\delta} \right)} + B_t \right) \sigma_t(\cdot) \\
&= \left(\sqrt{2\sigma \log \left(\det(I_t + K_t/\sigma)^{\frac{1}{2}} \right) - 2\sigma \log(\delta)} + B_t \right) \sigma_t(\cdot) \\
&= \left(\sqrt{\sigma \log \left(\det(I_t + K_t/\sigma) \right) - 2\sigma \log(\delta)} + B_t \right) \sigma_t(\cdot).
\end{aligned}$$

□

B DERIVATION OF THE RKHS NORM FORMULA

In this section, we derive the general formula of the RKHS norm, i.e.,

$$\|f\|_k^2 = \sum_{s=1}^{\infty} \sum_{t=1}^{\infty} \alpha_i \alpha_j k(x_s, x_t).$$

Let $f \in H_k$. Then, we can write

$$f = \sum_{t=1}^{\infty} \alpha_t k(\cdot, x_t)$$

In Hilbert spaces, the norm is given by the square root of the inner product of the function. Hence,

$$\|f\|_k^2 = \langle f, f \rangle_k,$$

where $\langle \cdot, \cdot \rangle_k$ denotes the inner product of two functions in the RKHS of kernel k. Therefore, we have

$$\begin{aligned}\|f\|_k^2 &= \langle \sum_{t=1}^{\infty} \alpha_t k(\cdot, x_t), \sum_{t=1}^{\infty} \alpha_t k(\cdot, x_t) \rangle_k \\ &= \sum_{t=1}^{\infty} \alpha_t k(\cdot, x_t) \sum_{s=1}^{\infty} \alpha_s k(\cdot, x_s) \\ &= \sum_{t=1}^{\infty} \sum_{s=1}^{\infty} \alpha_t \alpha_s k(\cdot, x_t) k(\cdot, x_s) \\ &= \sum_{t=1}^{\infty} \sum_{s=1}^{\infty} \alpha_t \alpha_s k(x_s, x_t),\end{aligned}$$

where the last equality follows from the reproducing property of reproducing kernel Hilbert spaces. The "center" points are the x_s (or x_t) points in this sum.

C ADDITIONAL FIGURE FOR THE INTRODUCTORY EXAMPLE

Figure 8 shows the effect of conducting safe BO with a too conservative upper bound on the RKHS norm.

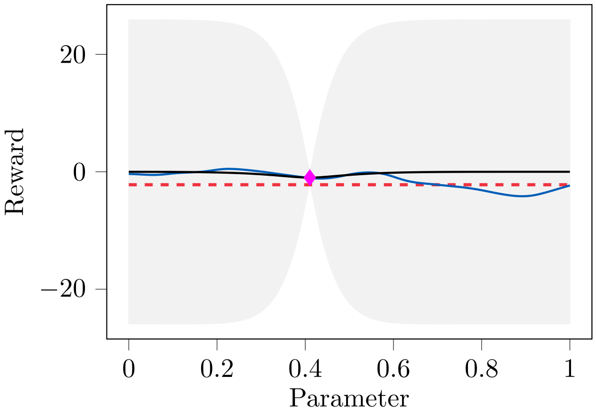

Figure 8: *Safe BO corresponding to Figure 1.* In this case, the guessed RKHS norm is five times the true RKHS norm, i.e., the RKHS norm is conservatively over-estimated. The safe BO algorithm cannot sample any parameter since none is safe with high probability. Hence, a conservative over-estimation of the RKHS norm is undesirable.

D 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 variance 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 (Hochreiter and Schmidhuber, 1997) 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; see Figure 9 for a schematic diagramm of the RNN.

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

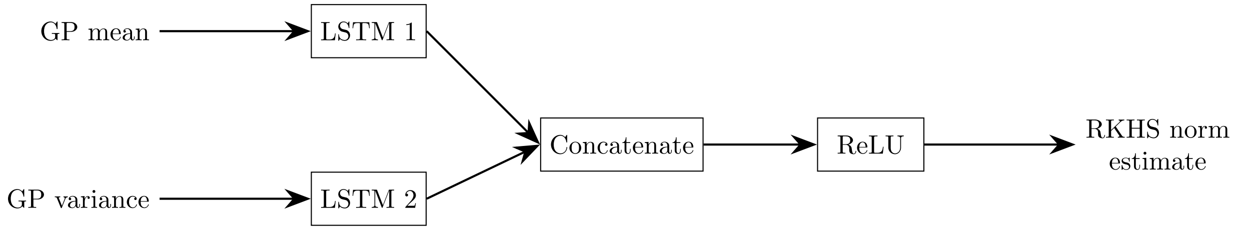

Figure 9: Schematic diagram of the used RNN.

$\delta = 10^{-2}$, $\kappa = 10^{-2}$, $\gamma = 10^{-1}$, $\Delta = 10^{-1}$, 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., Maddalena et al. (2021) for the computation of the RKHS norm of the interpolating function.

Performance The 10^3 functions g yield 280×10^3 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 min. We additionally preserved 20% of validation data. The root mean squared error on the validation data was approximately 5×10^{-3}.

Role of the RNN As mentioned in Section 3.1, the RNN merely provides an additional layer of conservatism on the RKHS norm over-estimation and does not influence the provided theoretical guarantees. However, it assists in accelerating Algorithm 4. In Algorithm 4, we first loop through all local cubes $\mathcal{C}_t$ and determine the most uncertain interesting parameter within each cube. Then, we conduct an experiment with the most uncertain interesting parameter among all local cubes. Note that we only require PAC bounds, i.e., guarantees on the RKHS norm over-estimation when conducting the experiment and not while looping through each local cube. Therefore, to accelerate Algorithm 4, we loop through the local cubes and determine the uncertainties using only the RKHS norm estimation of the RNN since we do not conduct a safety-critical experiment yet. Then, when having determined the most uncertain interesting parameter, we compute the PAC RKHS norm over-estimation and check whether the parameter remains safe with high probability before conducting the experiment.

E FURTHER ELABORATION ON ASSUMPTION 2

Assumption 2 holds if the ground truth f and the random RKHS functions $\rho_{t,j}$ are i.i.d. samples from the same—potentially unknown—probability space. In the following, we demystify the assumption and contrast it to the assumption made in SafeOpt, which is to assume that an upper bound on the RKHS norm is available a priori.

In concrete terms, our assumption restricts the complexity of the ground truth by requiring

$$f = \sum_{p=1}^{\hat{N}} \alpha_p k(x_p, \cdot).$$

Since we compute random RKHS functions of the form

$$\rho_{t,j} = \sum_{s=1}^{\hat{N}} \alpha_s k(x_s, \cdot), \quad t \geq 1, j \in \{1, \dots, \mathrm{m}\},$$

Assumption 2 holds if $\rho_{t,j}$ and f are i.i.d. samples from the same—potentially unknown—probability space. We construct the random RKHS functions by deterministically fixing the first $t \ll \hat{N}$ coefficients $\alpha_{i:t}$ and center points $x_{1:t}$ by the interpolation property subject to σ-sub-Gaussian measurement noise. The remaining coefficients and center points are i.i.d. samples from some probability space $(\Omega, \mathcal{F}, \nu)$. Therefore, Assumption 2 holds if

$$f = \sum_{p=1}^{\hat{N}} \alpha_p k(x_p, \cdot), \quad \alpha_p, x_p \begin{cases} \text{from interpolation property} & \text{if } p \in [1, t] \\ \text{i.i.d. samples from } (\Omega, \mathcal{F}, \nu) & \text{if } p \in [t+1, \hat{N}]. \end{cases} \tag{10}$$

In our experiments, we sample $\alpha_p \in [-\bar{\alpha}, \bar{\alpha}] = [-1, 1]$ and $x_s \in \mathcal{A} = [0, 1]$ from *uniform* distributions, allowing for a worst-case RKHS norm of up to 500. In comparison, SafeOptrequires

$$f = \sum_{p=1}^{\infty} \alpha_p k(x_p, \cdot), \quad \|f\|_k = \sqrt{\sum_{p=1}^{\infty} \sum_{s=1}^{\infty} \alpha_p \alpha_s k(x_p, x_s)} \overset{!}{\leq} B, \tag{11}$$

where B is the a priori upper bound on the RKHS norm of the ground truth f.

Essentially, both assumptions, i.e., (10) and (11), restrict the complexity of the ground truth by assuming sufficiently regular behavior. Regularity assumptions are necessary as considering arbitrarily complex functions would not lead to any practical bounds. In practice, we need to approximate the probability space $(\Omega, \mathcal{F}, \nu)$. Consequently, SafeOpt needs to approximate the probability space as well; the set of possible outcomes are the functions from that RKHS *subject to the RKHS norm condition* $\|f\|_k \leq B$. The probability distribution in this setting is nature's probability distribution, which is the classic frequentist setting. In contrast to SafeOpt, our assumption captures larger RKHS norms and systematically incorporates data instead of having a static a priori restriction on the functions.

We mention throughout the paper how, why, and when SafeOpt's assumption breaks; more importantly, when this a priori restriction of possible functions for the ground truth f leads to safety violations. To make our assumption work in practice, we choose $\hat{N}$ and $\bar{\alpha}$ in (10) large enough to cover a broad range of functions. Heuristically, it is sensible to restrict ourselves to a pre-RKHS setting (i.e., $\hat{N} < \infty$) since the bounded norm property of $\|f\|_k < \infty$ "implies that the coefficients α_p decay sufficiently fast as p increases" (Berkenkamp et al., 2023).

Contrasting Assumption 2 to Assumption 1 by Tokmak et al. (2024) Assumption 2 essentially states that the random RKHS functions and the reward function are i.i.d. samples from the same—potentially unknown—probability space. In comparison, Assumption 1 in Tokmak et al. (2024) requires that the RKHS norms of the random RKHS functions over-estimate the RKHS norm of the reward function in expectation. Hence, the connection between the random RKHS functions and the ground truth does not directly become obvious since it is unclear under which conditions on the ground truth and the random RKHS functions this assumption holds. In contrast, our assumption imposes a direct connection between the random RKHS functions and the ground truth.

F PROOFS

In this section, we provide the proofs of our theoretical contributions, which we presented in Section 3.2. For the reader's convenience, we will restate the mathematical claims before providing their proofs within each subsection.

F.1 Proof of Theorem 1

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 B_t returned by Algorithm 3. With confidence at least $1 - \kappa$, we have $B_t \geq \|f\|_k$ with probability at least $1 - \gamma$.*

We prove the theorem by following a (sampling-and-discarding) scenario approach (Campi and Garatti, 2011; Calafiore and Campi, 2006).

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

$$\begin{aligned} &\min_{B_t^* \in \mathbb{R}_{\geq B_t}} \quad B_t^* \\ &\text{subject to} \quad B_t^* \geq \|f\|_k. \end{aligned} \tag{12}$$

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 (12) is not solvable since $\|f\|_k$ is unknown. Hence, we formulate the optimization problem using the scenario approach (Calafiore and Campi, 2006) with m i.i.d. random RKHS functions $\rho_{t,j}$:

$$\begin{aligned} \min_{B_t^* \in \mathbb{R}_{\geq B_t}} \quad & B_t^* \\ \text{subject to} \quad & B_t^* \geq \|\rho_{t,j}\|_k \quad \forall j \in \{1, \dots, m\}. \end{aligned} \tag{13}$$

We can use a scenario approach (13) to tackle Problem (12) since the RKHS norms are i.i.d. random variables from the same probability space (Calafiore and Campi, 2006). Specifically, by solving (13), we obtain a solution that satisfies all m constraints, which, in return, yields a PAC solution for Problem (12). 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 (Campi and Garatti, 2011). To this end, we formulate the following scalar optimization problem:

$$\begin{aligned} \min_{B_t^* \in \mathbb{R}_{\geq B_t}} \quad & B_t^* \\ \text{subject to} \quad & B_t^* \geq \|\rho_{t,j}\|_k \quad \forall i \in \{1, \dots, m-r\} \\ & B_t^* < \|\rho_{t,j}\|_k \quad \forall j \in \{m-r+1, \dots, m\}, \end{aligned} \tag{14}$$

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 Theorem 2.1 by Campi and Garatti (2011). Consider the probability space $(\mathbb{R}_{\geq 0}, \mathcal{B}(\mathbb{R}_{\geq 0}), \mathbb{P})$. The probability space with m scenarios can be written as $(\mathbb{R}^m_{\geq 0}, \mathcal{B}(\mathbb{R}^m_{\geq 0}), \mathbb{P}^m)$, i.e., a classic *product probability space*, equivalent to the setting in Romao et al. (2022).

Before using Theorem 2.1 by Campi and Garatti (2011), 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. Moreover, Problem (14) is feasible for $m < \infty$ with a feasibility domain with nonempty interior and unique solution (C4). In fact, the solution of (14) is

$$B^\star_{t,m,r} = \max\{\|\rho_{t,m-r}\|_k, B_t\}, \tag{15}$$

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 $j = m-r+1, \dots, m$ in (14) almost surely. To this end, note that the RKHS norms are sorted in an ascending order and that $\|\rho_{t,j}\|_k \neq \|\rho_{t,i}\|_k, i,j \in \{1,\dots,m\}, i \neq j$ 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 and $\|\rho_{t,m-r}\|_k < \|\rho_{t,j}\|_k$, $\forall j \in \{m-r+1,\dots,m\}$ almost surely, the claim holds. Hence, we can use the result of Theorem 2.1 in Campi and Garatti (2011):

$$\begin{aligned}&\mathbb{P}^m\left[(\|\rho_{t,1}\|_k,\dots,\|\rho_{t,m}\|_k) \in \mathbb{R}^m_{\geq 0} : \mathbb{P}\left[\|f\|_k \in \mathbb{R}_{\geq 0} \colon B^\star_{t,m,r} \geq \|f\|_k\right] \geq 1-\gamma\right] \\ &\geq 1 - \sum_{i=0}^{r}\binom{m}{i}\gamma^i(1-\gamma)^{m-i}.\end{aligned} \tag{16}$$

Inequality (16) provides PAC bounds on the constraint satisfaction for any unknown random variable from the same probability space. Therefore, it probabilistically quantifies the constraint satisfaction of the optimal solution of (14) with respect to the unsolvable optimization problem (12), 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

$$\mathbb{P}^m\left[(\|\rho_{t,1}\|_k,\dots,\|\rho_{t,m}\|_k) \in \mathbb{R}^m_{\geq 0} : \mathbb{P}\left[\|f\|_k \in \mathbb{R}_{\geq 0} \colon B_t \geq \|f\|_k\right] \geq 1-\gamma\right] \geq 1-\kappa, \tag{17}$$

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 (14). Conditions (C1)-(C4) are satisfied equivalently to Case I, and Condition (C5) holds trivially since $r = 0$. The optimal solution of Problem (14) 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

$$\begin{aligned}\sum_{i=0}^{0}\binom{m}{i}\gamma^i(1-\gamma)^{m-i} &\leq \sum_{i=0}^{1}\binom{m}{i}\gamma^i(1-\gamma)^{m-i} \\ &= (1-\gamma)^{m-1}(1+\gamma(m-1)) \\ &\leq \kappa,\end{aligned} \tag{18}$$

we can directly obtain PAC bounds for the optimal solution of the sampling-and-discarding scenario approach (14) with $r = 0$. Namely,

$$\begin{aligned}&\mathbb{P}^m\left[(\|\rho_{t,1}\|_k,\dots,\|\rho_{t,m}\|_k) \in \mathbb{R}^m_{\geq 0} : \mathbb{P}\left[\|f\|_k \in \mathbb{R}_{\geq 0} \colon B_t \geq \|f\|_k\right] \geq 1-\gamma\right] \\ &\geq 1 - \sum_{i=0}^{0}\binom{m}{i}\gamma^i(1-\gamma)^{m-i} \overset{(18)}{\geq} 1-\kappa,\end{aligned}$$

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, 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. □

F.2 Proof of Corollary 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 confidence at least $1-\kappa$, B_t over-estimates the ground truth RKHS norm $\|f\|_k$ jointly for all iterations $t \geq 1$ with probability at least $1-\gamma$.*

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

$$B_t \leq B_{t-1} \leq \ldots \leq B_1 \quad \forall t \geq 1. \tag{19}$$

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 (19), $\mathbb{E}[B_t] \leq B_1 < \infty$.[3] Moreover,

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

follows from (19), i.e., $\{B_t\}_{t=1}^T$ is a supermartingale with respect to the filtration $\{\mathfrak{F}_t\}_{t=1}^T$ (Durrett, 2019, Section 4.2). Therefore, we can use a stopping-time construction for (super)martingales as done in Theorem 1 by Abbasi-Yadkori et al. (2011) and Theorem 1 by Chowdhury and Gopalan (2017).

Let us define the bad event

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

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

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

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

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

we have

$$\begin{aligned} \mathbb{P}[\cup_{t \geq 1} \mathscr{B}_t] &= \mathbb{P}[\tau' < \infty] \\ &= \mathbb{P}[B_t < \|f\|_k, \tau' < \infty] \\ &\leq \mathbb{P}[B_t < \|f\|_k]. \end{aligned} \tag{20}$$

In Theorem 1, we proved that

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

for any (fixed) $t \geq 1$. Therefore,

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

for any (fixed) $t \geq 1$, which with (20) implies that the statement in Theorem 1 now holds holds *jointly* for all $t \geq 1$. That is, lifting the statement to hold jointly for all iterations is to upper-bound the probability that the bad event $\mathscr{B}_t$ happens, which we do in (20). This probability is upper-bounded by the probability of under-estimating the RKHS norm. In words, with confidence at least $1-\kappa$, the RKHS norm over-estimation holds jointly for all iterations with probability at least $1-\gamma$, where the confidence and probability are stated with respect to the probability measures $\mathbb{P}$ and $\mathbb{P}^m$, respectively. □

[3] Note that the expected value is with respect to the probability space $(\mathbb{R}_{\geq 0}, \mathcal{B}(\mathbb{R}_{\geq 0}), \mathbb{P})$ (i.e., with respect to the probability measure $\mathbb{P}$) since the random variable B_t is defined on that probability space.

F.3 Proof of Theorem 2

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 (4) with any $\delta \in (0,1)$ and $C_t \coloneqq C_{t-1} \cap Q_t$ with $C_0 = \mathbb{R}$. Then, with confidence 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 at least $(1-\gamma)(1-\delta)$.*

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

$\mathfrak{C}_t$: It holds that $f(a) \in C_t(a)$ jointly for all $a \in \mathcal{A}$ and for all $t \geq 1$.
$\mathcal{E}_t$: It holds that $\|\epsilon_{1:t}\|_{((K_t+\sigma I_t)^{-1}+I_t)^{-1}} \leq 2\sigma^2 \ln\left(\frac{\sqrt{\det(1+\sigma)I_t+K_t}}{\delta}\right)$ jointly for all $t \geq 1$ and for any $\delta \in (0,1)$.
$\mathfrak{Q}_t$: It holds that $f(a) \in Q_t(a)$ jointly for all $a \in \mathcal{A}$ and for all $t \geq 1$.
$\mathfrak{B}_t$: It holds that $B_t \geq \|f\|_k$ jointly for all $t \geq 1$.

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

The challenge in this proof is that state-of-the-art confidence intervals from e.g., Abbasi-Yadkori (2013) or Chowdhury and Gopalan (2017) consider the RKHS norm as a *deterministic* object and, therefore, only have the sub-Gaussian measurement noise as the source of stochasticity. In contrast, we over-estimate the RKHS norm, thus making it a random variable. Therefore, we have two sources of uncertainty that are defined on two distinct probability spaces.

Uncertainty *(i)*. From Corollary 1, we have that

$$\mathbb{P}_1^m\left[(\|\rho_{t,1}\|_k,\dots,\|\rho_{t,m}\|_k) \in \mathbb{R}^m_{\geq 0} : \mathbb{P}_1\left[\|f\|_k \in \mathbb{R}_{\geq 0} \colon B_t \geq \|f\|_k\right] \geq 1-\gamma\right] \geq 1-\kappa.$$

jointly for all $t \geq 1$, i.e.,

$$\mathbb{P}_1^m\left[(\|\rho_{t,1}\|_k,\dots,\|\rho_{t,m}\|_k) \in \mathbb{R}^m_{\geq 0} : \mathbb{P}_1\left[\mathfrak{B}_t\right] \geq 1-\gamma\right] \geq 1-\kappa.$$

The bounds are derived with respect to the inner probability space $(\mathbb{R}_{\geq 0}, \mathcal{B}(\mathbb{R}_{\geq 0}), \mathbb{P}_1)$ and the outer product probability space $(\mathbb{R}^m_{\geq 0}, \mathcal{B}(\mathbb{R}^m_{\geq 0}), \mathbb{P}_1^m)$. The inner probability measure $\mathbb{P}_1$ quantifies the uncertainty on the *hypothesis* that the RKHS norm over-estimation is correct, while the outer probability measure $\mathbb{P}_1^m$ quantifies the *sampling-based* uncertainty.[4]

We map the (inner) probability space to a simpler and more interpretable but for our needs equivalent probability space. Instead of working on the sample space $\mathbb{R}_{\geq 0}$, we work with the introduced events $\mathfrak{B}_t$ and $\mathfrak{B}_t^{\mathrm{C}}$. Note that the event-based sample space and the σ-algebra are instances from the original sample space and σ-algebra. The events show a more interpretable version of the original probability space. However, since the events are equivalently represented within the old and new settings, we preserve the original probability measure $\mathbb{P}_1$ and obtain the probability space

$$(\{\mathfrak{B}, \mathfrak{B}^{\mathrm{C}}\}, 2^{\{\mathfrak{B}, \mathfrak{B}^{\mathrm{C}}\}}, \mathbb{P}_1),$$

with $2^{\{\mathfrak{B}, \mathfrak{B}^{\mathrm{C}}\}} \coloneqq \{\emptyset, \mathfrak{B}, \mathfrak{B}^{\mathrm{C}}, \{\mathfrak{B}, \mathfrak{B}^{\mathrm{C}}\}\}$, i.e., the σ-algebra is the power set of the sample space. In this discrete σ-algebra, the probability measure is given by the tabular mapping

$$\begin{aligned}\mathbb{P}_1[\{\mathfrak{B}, \mathfrak{B}^{\mathrm{C}}\}] &= 1\\ \mathbb{P}_1[\emptyset] &= 0\\ \mathbb{P}_1[\mathfrak{B}] &= 1-\gamma\\ \mathbb{P}_1[\mathfrak{B}^{\mathrm{C}}] &= \gamma,\end{aligned}$$

where the first two results follow from the definition of valid probability measures, the third equality follows from Corollary 1, while the final equality follows from the fact that $\mathbb{P}_1[\{\mathfrak{B}, \mathfrak{B}^{\mathrm{C}}\}] = \mathbb{P}_1[\mathfrak{B}] + \mathbb{P}_1[\mathfrak{B}^{\mathrm{C}}]$ since $\mathfrak{B}$ and $\mathfrak{B}^{\mathrm{C}}$ are disjoint by construction.

[4] The random variable B_t is computed using a sampling-based approach by sampling m i.i.d. random RKHS functions; see Theorem 1. Since the generation of this "training set" is random, the resulting hypothesis is endowed with additional uncertainty, which requires us to introduce the outer layer of probability.

Remark 3 (Source of Uncertainty *(i)*)**.** *The uncertainty arises from the randomness of the random RKHS functions. As described in Section 3.1, the first t center points and coefficients are* deterministic *and set given the collected data points by the interpolating property subject to σ-sub-Gaussian noise. The randomness is solely introduced by sampling the tail coefficients and tail center points from, e.g., uniform distributions. Therefore, this uncertainty is purely epistemic.*

Uncertainty *(ii)*. From the works of Chowdhury and Gopalan (2017) and Abbasi-Yadkori (2013), we have probabilistic confidence intervals with a deterministic upper bound on the RKHS norm. Hence, the uncertainty arises solely from the sub-Gaussian measurement noise, i.e., from the probability of the occurrence of event $\mathcal{E}_t$. We form an equivalent event-based probability space transformation based on event $\mathcal{E}_t$ as we did for Uncertainty *(i)* instead of working on the original probability space. The event-based probability space is given by

$$((\mathcal{E}_t, \mathcal{E}_t^{\mathrm{C}}), 2^{\{\mathcal{E}_t, \mathcal{E}_t^{\mathrm{C}}\}}, \mathbb{P}_2).$$

The σ-algebra is again the power set of the sample space, i.e., $2^{\{\mathcal{E}_t, \mathcal{E}_t^{\mathfrak{C}}\}} = \{\{\mathcal{E}_t, \mathcal{E}_t^{\mathrm{C}}\}, \mathcal{E}_t, \mathcal{E}_t^{\mathrm{C}}, \emptyset\}$. We take the same probability measure as in the original probability space and write the probability measure as the tabular mapping

$$\begin{aligned}
\mathbb{P}_2[\{\mathcal{E}_t, \mathcal{E}_t^{\mathrm{C}}\}] &= 1 \\
\mathbb{P}_2[\emptyset] &= 0 \\
\mathbb{P}_2[\mathcal{E}_t] &= 1 - \delta \\
\mathbb{P}_2[\mathcal{E}_t^{\mathrm{C}}] &= \delta.
\end{aligned}$$

The fact that $\mathbb{P}_2[\mathcal{E}_t] = 1 - \delta$ is derived in Theorem 1 by Chowdhury and Gopalan (2017) and Theorem 3.4 by Abbasi-Yadkori (2013). The uncertainty of this event is purely aleatoric. It arises from applying Markov's inequality (Durrett, 2019, Theorem 1.6.4.) to probabilistically bound the norm of the accumulated measurement noise with respect to a positive definite matrix.

Since we want to combine Uncertainties *(i)* and *(ii)*, we extend $\mathbb{P}_2[\mathcal{E}_t]$ with the outer probability $\mathbb{P}_1^m$. That is, we include the sampling-based probability uncertainty on the training set of the random RKHS functions into the uncertainty of the measurement noise of conducting experiments. This extension is trivial and purely artificial since the generation of the random RKHS functions does not influence the measurement noise. Therefore, we can state

$$\mathbb{P}_1^m \left[(\|\rho_{t,1}\|_k, \dots, \|\rho_{t,m}\|_k) \in \mathbb{R}_{\geq 0}^m : \mathbb{P}_2[\mathcal{E}_t] \geq 1 - \delta \right] = 1.$$

Constructing a product probability space. Uncertainty *(i)* is purely epistemic and Uncertainty *(ii)* is purely aleatoric and both uncertainties are independent from each other. Therefore, we can create a *product probability space* between both individual probability spaces to quantify the uncertainty on the confidence interval, i.e., on events $\mathfrak{Q}_t$ and $\mathfrak{C}_t$ while treating the RKHS norm as a random variable.

First, we create a the unique product probability measure as described in, e.g., Theorem 1.7.1 by Durrett (2019). The unique probability measure of the product probability space is given by

$$\mathbb{P}[(\mathfrak{B}_t, \mathcal{E}_t)] = \mathbb{P}_1[\mathfrak{B}_t] \cdot \mathbb{P}_2[\mathcal{E}_t],$$

where we denote by $\mathbb{P}[(\cdot, \cdot)]$ the probability measure of the product probability space,[5] which maps a tuple of orthogonal random variables to the probability of joint occurrence. Since

$$\mathbb{P}[(\mathfrak{B}_t, \mathcal{E}_t)] = \mathbb{P}_1[\mathfrak{B}_t] \cdot \mathbb{P}_2[\mathcal{E}_t],$$

we have that

$$\mathbb{P}[(\mathfrak{B}_t, \mathcal{E}_t)] = (1 - \gamma)(1 - \delta).$$

[5] The product probability space is naturally given by the triple $((\mathfrak{B}_t, \mathfrak{B}_t^{\mathrm{C}}) \times (\mathcal{E}_t, \mathcal{E}_t), 2^{\{\mathfrak{B}_t, \mathfrak{B}_t^{\mathrm{C}}\} \times (\mathcal{E}_t, \mathcal{E}_t^{\mathrm{C}})}, \mathbb{P})$.

Moreover, we embed the uncertainty of the hypothesis into the stochasticity of the sampling process of the random RKHS functions by enveloping the inner probabilistic statement with the outer probability given by the measure $\mathbb{P}_1^m$. Hence, in conclusion, we can write

$$\mathbb{P}_1^m\left[(\|\rho_{t,1}\|_k,\ldots,\|\rho_{t,m}\|_k)\in\mathbb{R}^m_{\geq 0}:\mathbb{P}\left[(\mathfrak{B}_t,\mathcal{E}_t)\right]\geq(1-\gamma)(1-\delta)\right]\geq(1-\kappa)\cdot 1=1-\kappa$$

Now, note that the ground truth $f(a)$ lies within the confidence interval $Q_t(a)$ if *(i)* the RKHS norm over-estimation and *(ii)* the bound on the accumulated noise hold, i.e., if the event tuple $(\mathfrak{B}_t,\mathcal{E}_t)$ holds. Therefore, event $\mathfrak{Q}_t$ is *equivalent* to event $(\mathfrak{B}_t,\mathcal{E}_t)$ and we can write

$$\mathbb{P}_1^m\left[(\|\rho_{t,1}\|_k,\ldots,\|\rho_{t,m}\|_k)\in\mathbb{R}^m_{\geq 0}:\mathbb{P}\left[\mathfrak{Q}_t\right]\geq(1-\gamma)(1-\delta)\right]\geq 1-\kappa.$$

In words, with confidence at least $1-\kappa$, the hypothesis that the ground truth lies within the confidence intervals with with the measurement noise *and* the RKHS norm as random variables holds with probability at least $(1-\gamma)(1-\delta)$. Finally, from Corollary 7.1 by Berkenkamp et al. (2023), we have that

$$\mathbb{P}[\mathfrak{C}_t]\equiv\mathbb{P}[\mathfrak{Q}_t]$$

and, therefore,

$$\mathbb{P}_1^m\left[(\|\rho_{t,1}\|_k,\ldots,\|\rho_{t,m}\|_k)\in\mathbb{R}^m_{\geq 0}:\mathbb{P}\left[\mathfrak{C}_t\right]\geq(1-\gamma)(1-\delta)\right]\geq 1-\kappa.$$

The confidence is stated with respect to probability measure $\mathbb{P}_1^m$ that comprises the stochasticity of the random RKHS function generation, whereas the probability of the hypothesis is stated with respect to the product probability measure $\mathbb{P}$. □

F.4 Proof of Theorem 3

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

The proof is similar to the proofs of Theorem 1 and Lemma 11 by Sui et al. (2015). Specifically, we prove that we remain safe with high probability when only sampling within the set of safe samples.

SafeOpt-like algorithms start with an initial safe set and extend the safe set by (probabilistically) lower-bounding the function values of inputs on the domain based on information of inputs that are already classified as safe. This interpretation naturally requires a notion of continuity and regularity to infer the behavior of inputs based on the behavior of neighboring inputs. Therefore, SafeOpt and many SafeOpt-like algorithms require the Lipschitz constant as an additional parameter next to the RKHS norm to execute the algorithm. In contrast, we present a Lipschitz-like continuity for RKHS functions for which we use the (semi)metric (5).

Lemma 1 (RKHS-induced continuity)**.** *(Fiedler, 2023, Proposition 3.1) Let all conditions in Theorem 3 hold and let B_t be returned by Algorithm 3. Then, jointly for all $a,a'\in\mathcal{A}$ and for all $t\geq 1$, with confidence at least $1-\kappa$,*

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

with probability at least $1-\gamma$.[6]

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

$$\begin{aligned}|f(a)-f(a')|&=|\langle f,k(a,\cdot)-k(a',\cdot)\rangle_k| && \text{(Steinwart and Christmann, 2008, Definition 4.18)}\\&\leq\|f\|_k\sqrt{k(a,a)-k(a',a)-k(a,a')+k(a',a')} && \text{Cauchy-Schwarz inequality}\\&=\|f\|_k d_k(a,a') && \text{(Semi)metric (5)}\\&\leq B_t d_k(a,a'), && \text{Corollary 1}\end{aligned}$$

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. □

[6] The confidence is stated with respect to the probability measure $\mathbb{P}_1^m$ whereas the probability is stated with respect to the probability measure $\mathbb{P}_1$.

For each iteration $t \geq 1$, we are only allowed to sample within the safe set S_t (6). 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 at least* $1-\kappa$*,*

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

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

Proof. The lemma is akin to Lemma 11 by Sui et al. (2015). However, we replace the assumption of knowing a true upper bound on the RKHS norm $\|f\|_k$ with the PAC RKHS norm over-estimation received by Algorithm 3. Furthermore, in contrast to Sui et al. (2015), 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:
Σ_t: It holds that $f(a) \geq h$ jointly for all $a \in S_t$ and for all $t \geq 1$.
$\mathfrak{C}_t$: It holds that $f(a) \in C_t(a)$ jointly for all $a \in \mathcal{A}$ and for all $t \geq 1$.
$\mathcal{E}_t$: It holds that $\|\epsilon_{1:t}\|_{((K_t+\sigma I_t)^{-1}+I_t)^{-1}} \leq 2\sigma^2 \ln\left(\frac{\sqrt{\det(1+\sigma)I_t+K_t}}{\delta}\right)$ jointly for all $t \geq 1$ and for any $\delta \in (0,1)$.
$\mathfrak{B}_t$: It holds that $B_t \geq \|f\|_k$ jointly for all $t \geq 1$.
$\mathfrak{L}_t$: It holds that $|f(a) - f(a')| \leq B_t d_k(a,a')$ jointly for all $a \in \mathcal{A}$ and for all $t \geq 1$.

First, we project the statement in Lemma 1 onto the product probability space before providing a lower bound on the occurrence of the event Σ_t. From Lemma 1, we have that

$$\mathbb{P}_1^m\left[(\|\rho_{t,1}\|_k, \dots, \|\rho_{t,m}\|_k) \in \mathbb{R}^m_{\geq 0} : \mathbb{P}_1\left[\mathfrak{L}_t\right] \geq 1-\gamma\right] \geq 1-\kappa.$$

We now embed the inner probability from probability measure $\mathbb{P}_1$ into the product probability measure $\mathbb{P}$. Note that event $\mathfrak{L}_t$ (on the probability space governed by the probability measure $\mathbb{P}_1$) solely depends on the correctness of the RKHS norm over-estimation, i.e., on the occurrence of event $\mathfrak{B}_t$. Therefore, with respect to the product probability measure $\mathbb{P}$, event $\mathfrak{L}_t$ is *equivalent* to the union of the events $(\mathfrak{B}_t, \mathcal{E}_t) \cup (\mathfrak{B}_t, \mathcal{E}_t^{\mathrm{C}})$ since it solely depends on the correctness of the RKHS norm over-estimation. Therefore, we can write

$$\begin{aligned}\mathbb{P}_1\left[\mathfrak{L}_t\right] \equiv \mathbb{P}[(\mathfrak{B}_t, \mathcal{E}_t^{\mathrm{C}}) \cup (\mathfrak{B}_t, \mathcal{E}_t^{\mathrm{C}})] &= \mathbb{P}[(\mathfrak{B}_t, \mathcal{E}_t)] + \mathbb{P}[(\mathfrak{B}_t, \mathcal{E}_t^{\mathrm{C}})] \\ &= (1-\gamma)(1-\delta) + (1-\gamma)\delta \\ &= 1-\gamma,\end{aligned}$$

i.e., we can write $\mathfrak{L}_t \coloneqq (\mathfrak{B}_t, \mathcal{E}_t) \cup (\mathfrak{B}_t, \mathcal{E}_t^{\mathrm{C}})$. Therefore, the probability that the continuity statement from Lemma 1 holds is given by

$$\mathbb{P}_1^m\left[(\|\rho_{t,1}\|_k, \dots, \|\rho_{t,m}\|_k) \in \mathbb{R}^m_{\geq 0} : \mathbb{P}\left[\mathfrak{L}_t\right] \geq 1-\gamma\right] \geq (1-\kappa).$$

Remark 4 (Introduced events and their role in the product probability space)**.** *We did not explicitly introduce* $\mathfrak{L}_t$ *into the* σ*-algebra of the product probability space, which is the power set of the sample set* $(\mathfrak{B}_t, \mathfrak{B}_t^{\mathrm{C}}) \times (\mathcal{E}_t, \mathcal{E}_t^{\mathrm{C}})$*. However, we showed that* $\mathfrak{L}_t$ *can be rewritten as* $(\mathfrak{B}_t, \mathcal{E}_t) \cup (\mathfrak{B}_t, \mathcal{E}_t^{\mathrm{C}})$*, thus making it an explicit a member of the* σ*-algebra of the product probability space. Clearly,* $(\mathfrak{B}_t, \mathcal{E}_t) \cup (\mathfrak{B}_t, \mathcal{E}_t^{\mathrm{C}}) = \mathfrak{B}_t$*, i.e.,* $\mathfrak{L}_t \equiv \mathfrak{B}_t$*. This already became evident in the proof of Lemma 1, where the stochasticity of the continuity statement was solely introduced by the stochasticity on the RKHS norm over-estimation.*

We now move on to lower-bounding the probability of occurrence of event Σ_t, which clearly proves the lemma. Equivalent to the proof of Lemma 11 in Sui et al. (2015), we prove the lemma by induction.

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 $f(a') \geq h,\ \forall a' \in S_t$. By the definition of the safe set (6), $\forall a' \in S_t$, $\exists a \in S_{t-1}$ such that

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

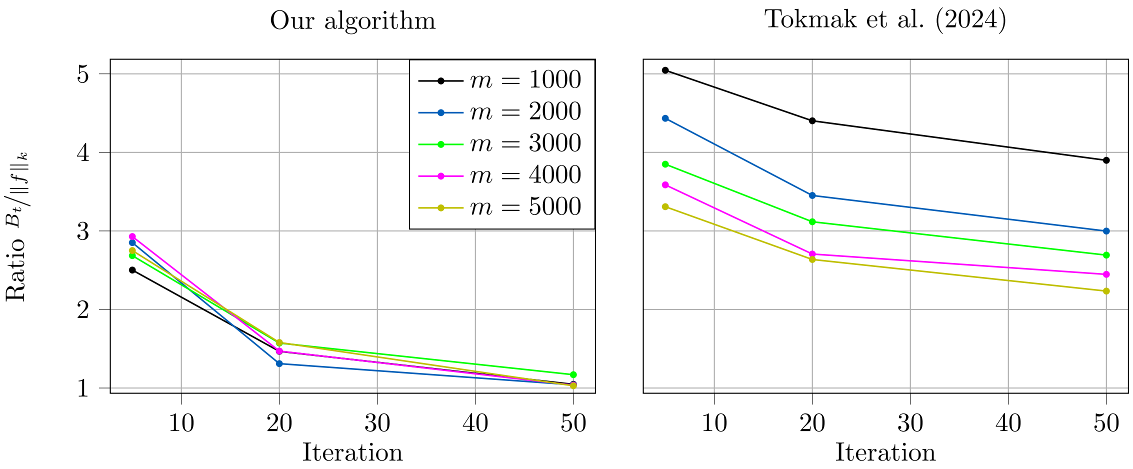

Figure 10: Tightness of the RKHS norm over-estimation of our algorithm (left) compared to Tokmak et al. (2024) (right) when computing m random RKHS functions.

holds deterministically. Moreover, we have shown in Theorem 2, that

$$\mathbb{P}_1^m \left[(\|\rho_{t,1}\|_k, \dots, \|\rho_{t,m}\|_k) \in \mathbb{R}^m_{\geq 0} : \mathbb{P}\left[\mathfrak{C}_t\right] \geq (1-\gamma)(1-\delta) \right] \geq 1-\kappa,$$

which implies that[7]

$$\mathbb{P}_1^m \left[(\|\rho_{t,1}\|_k, \dots, \|\rho_{t,m}\|_k) \in \mathbb{R}^m_{\geq 0} : \mathbb{P}\left[f(a) \geq \ell_t(a)\right] \geq (1-\gamma)(1-\delta) \right] \geq 1-\kappa.$$

Therefore,

$$\mathbb{P}_1^m \left[(\|\rho_{t,1}\|_k, \dots, \|\rho_{t,m}\|_k) \in \mathbb{R}^m_{\geq 0} : \mathbb{P}\left[h \leq f(a) - B_t d_k(a, a')\right] \geq (1-\gamma)(1-\delta) \right] \geq 1-\kappa.$$

Finally, observe that event $\mathfrak{C}_t$ implies event $\mathfrak{L}_t$ (see Remark 4). Therefore,

$$\mathbb{P}_1^m \left[(\|\rho_{t,1}\|_k, \dots, \|\rho_{t,m}\|_k) \in \mathbb{R}^m_{\geq 0} : \mathbb{P}\left[h \leq f(a')\right] \geq (1-\gamma)(1-\delta) \right] \geq 1-\kappa$$

by Lemma 1.

□

Theorem 3 follows directly from Lemma 2. □

G RKHS NORM OVER-ESTIMATION TIGHTNESS COMPARED TO Tokmak et al. (2024)

Figure 10 shows the tightness of the RKHS norm over-estimation by showing the ratio $B_t/\|f\|_k$ over the number of iterations. We see that the RKHS norm over-estimation using the scenario approach (left sub-figure, our work) yields *significantly* tighter over-estimations than the work by Tokmak et al. (2024), who directly use Hoeffding's inequality to obtain PAC bounds.

H ADAPTIVE NOTION OF LOCALITY

In Section 3.3, we explain the motivation behind exploiting locality to reduce conservatism and, thus, improve exploration. Figure 11 shows a toy example in which working with the *global* RKHS norm would result in unnecessarily conservative exploration in most parts of the domain. This is because the sub-domain in the center has the largest slope and, thus, mainly contributes to increasing the (global) RKHS norm, whereas the

[7] The implication follows from the fact that $\min C_t(a) =: \ell_t(a)$.

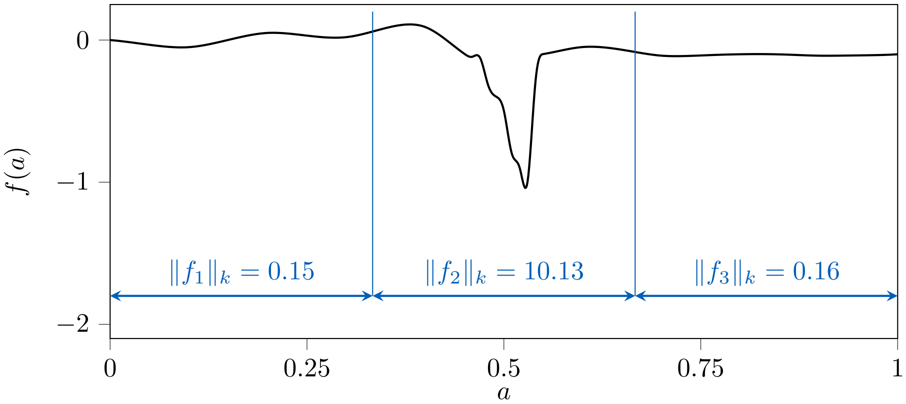

Figure 11: *General idea of locality.* We introduce a local interpretation of the RKHS norm and local sub-domains to exploit local “smoothness” and, thus, work with smaller RKHS norms on sub-domains with smaller slopes.

sub-domains on the right- and left-hand side have *significantly* smaller slopes and local RKHS norms. These locally smaller RKHS norms naturally yield tighter confidence intervals when defining them on these local sub-domains.

Now, we contrast our *adaptive* notion of locality to the locality introduced in Tokmak et al. (2024). As mentioned in Section 3.3, we define sub-domains as local cubes around each sample $a \in a_{1:t}$. The size of these local cubes and the number of local cubes around each sample are hyperparameters. In addition to these sub-domains, we preserve the global domain.

In contrast, Tokmak et al. (2024) use *(i)* the global domain, *(ii)* the convex hull of the samples, *(iii)* some intermediate domain between the convex hull and the global domain. A weakness of this approach is that it requires at least two samples to start exploration since the convex hull of a singleton set is purposeless in the considered setting. Furthermore, let $S_0 = \{0, 1\}$ be the set of initial safe samples on the global domain $\mathcal{A} = [0, 1]$. Then, the locality introduced by Tokmak et al. (2024) would again be impractical since the convex hull of samples is equal to the global domain, resulting in global exploration without a notion of locality. This design choice introduces limitations on the initial samples, restricting practical applicability.

In our paper, singleton safe sets and points far apart do not negatively influence the notion of locality. Furthermore, we can additionally influence the size of the domain with the hyperparameter Δ. This hyperparameter is a major difference from Tokmak et al. (2024) and the main reason why our algorithm is significantly more scalable than SafeOpt. Consider, e.g., the six-dimensional half-cheetah environment, where we set $\Delta = 0.05$. In SafeOpt, or in the unfavorable local setting of Tokmak et al. (2024), we would revert to $\Delta = 1$, which would give a Euclidean distance of $\mathcal{O}(10^{-1})$ between the samples, whereas the distance in our case can be reduced by a factor of 200. This improves scalability, thus allowing for exploration in higher dimensions.

I REQUIRED COMPUTATION TIME FOR THE SCENARIO APPROACH

We investigate the required computation time for the scenario approach. Additionally, we comment on the influence on the total computation time when integrating the scenario approach in SafeOpt-type algorithms.

Figure 12 shows the computation time of the scenario approach over the number of iterations for different m. The required time expectedly increases with the number of constraints m, whereas a significant dependence of the computation time on the iteration cannot be observed. This is in huge contrast to SafeOpt. The computation time of SafeOpt is investigated in Figure 2 by Baumann et al. (2023), where a significant increase is observable with the number of iterations. Crucially, the influence of the scenario approach on the total computation time is insignificant; especially because our algorithm, akin to SafeOpt, works in an episodic setting, where we do not have to meet real-time requirements.

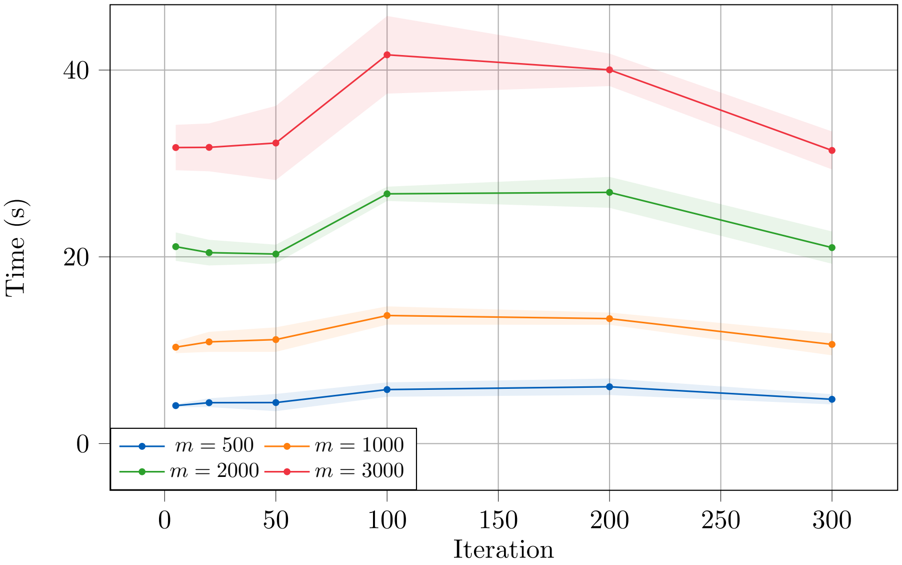

Figure 12: *Computation time for the scenario approach with m random RKHS functions.* The computation time is independent of the iteration and, hence, the amount of collected data.

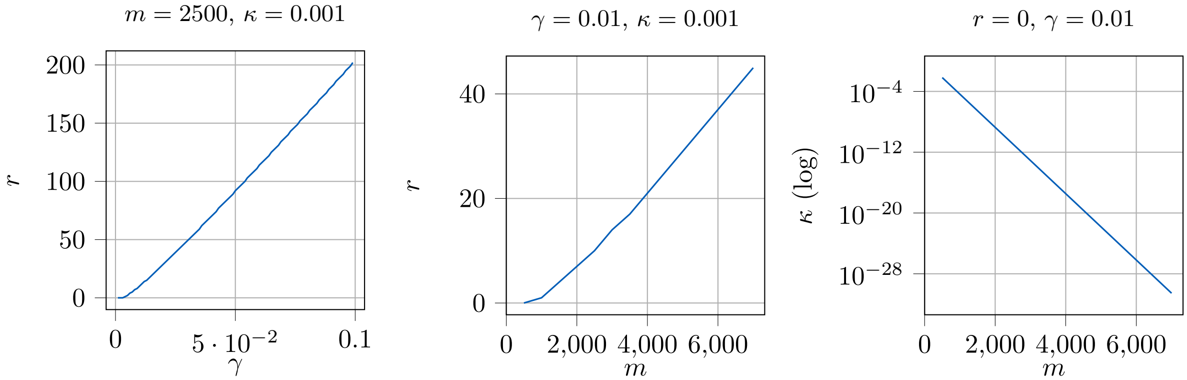

Figure 13: Influence of the hyperparameters of the scenario approach.

J ABLATION STUDY FOR THE SCENARIO APPROACH

Figure 13 shows an ablation study of the hyperparameters γ, m, r, and κ for the scenario approach. *Left:* The number of constraints r that we can discard grows linearly with the decrease of the parameter γ when keeping m and κ constant. *Center:* The number of constraints r that we can discard grows linearly with the number of total constraints m when keeping γ and κ constant. *Right:* The confidence parameter κ decreases exponentially with the number of total constraints m when keeping γ and r constant.

K ABLATION STUDY USING ALGORITHM 4 WITH DIFFERENT KERNELS

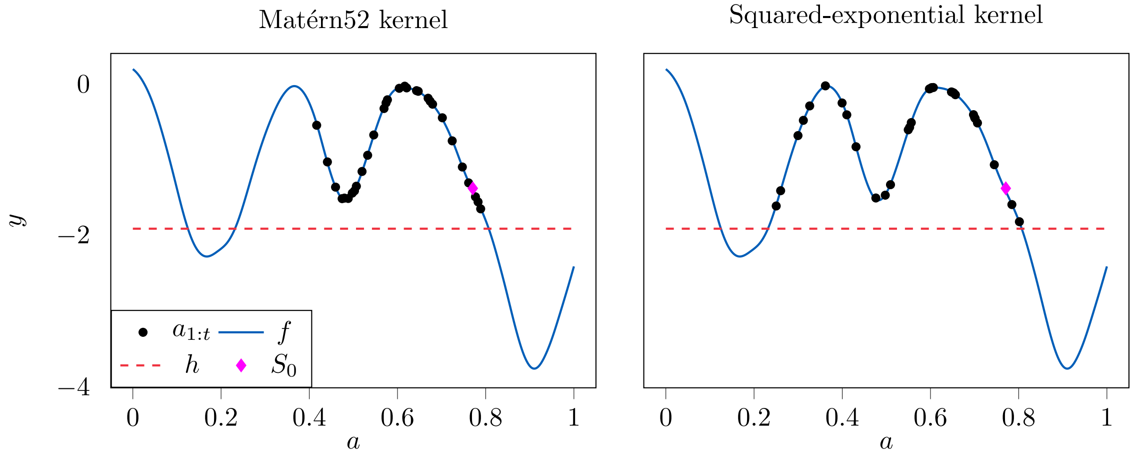

Figure 14: *Performance of our safe BO algorithm using different kernels.* Even with a misspecified kernel, our algorithm explores the domain safely.

In Figure 14, we perform an ablation study by conducting the numerical experiment from Section 5 using Algorithm 4 with different kernels. Hence, the kernel used in Algorithm 4 differs from the one with which the reward function is created. Although we cannot provide any theoretical guarantees for this setting, Figure 14 shows the successful deployment of our algorithm in a setting in which the kernel is misspecified.

L ABLATION STUDY USING ALGORITHM 4 WITH DIFFERENT LOCALITY PARAMETERS

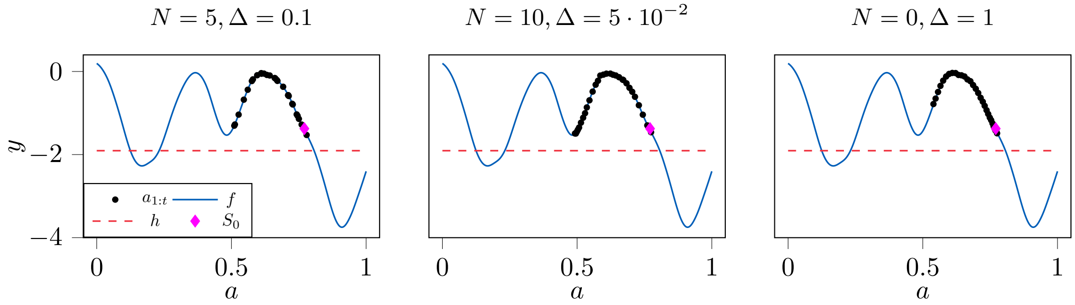

Figure 15: Performance of our safe BO algorithm using different locality parameters N and Δ.

In Figure 15, we perform an ablation study by conducting the numerical experiment from Section 5 using Algorithm 4 with different locality parameters N and Δ. The left sub-figure is created using the original locality settings described in Section 5, i.e., it is identical to the left sub-figure in Figure 5. The sub-figure in the center is created using more and smaller local cubes. Therefore, Algorithm 4 has more options for local cubes to iterate through, which (slightly) improves exploration behavior. The right sub-figure results from executing

Algorithm 4 without local cubes, i.e., by only iterating through the global domain $\mathcal{A}$. Therefore, we naturally recover Algorithm 1. As expected and stated in Section 3.3, we observe inferior exploration compared to the setting that actively exploits locality.

M 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 (Haarnoja et al., 2018) in various OpenAI Gym environments (Brockman et al., 2016), in particular, the mountain car, the cart-pole system, the swimmer, the lunar lander, the half-cheetah, and the ant. 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 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 behind 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 5×10^2 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 5×10^2 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.

Ant (8D) We set $N = 10$, $\Delta = 5 \times 10^{-2}$, and discretize the environment with 5 points per dimension. For the imitated real experiments, we change the thickness of the leg joint from 0.08 to 0.18. The goal is to ensure forward movement without falling; any fall was considered unsafe.

N HARDWARE EXPERIMENT

We conducted the hardware experiment on an Ubuntu laptop with 32 GB RAM and an Intel Core i7-12700H processor. Figure 16 shows the setup of the Furuta pendulum.

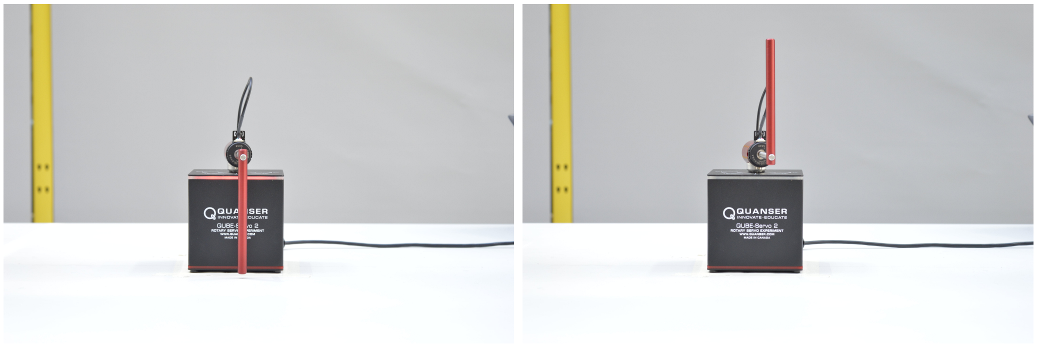

Figure 16: *Hardware setup.* The Furuta pendulum starts from a downward position (left) and is swung upright. Then, we use a state-feedback controller to balance the pole (right).