Safely Learning Controlled Stochastic Dynamics

Luc Brogat-Motte
Laboratoire des Signaux et Systèmes, CNRS, CentraleSupélec
Université Paris-Saclay, Gif-sur-Yvette, France
Istituto Italiano di Tecnologia, Genoa, Italy
luc.brogatmotte@iit.it

Alessandro Rudi
SDA Bocconi, Bocconi University, Milan, Italy
alessandro.rudi@sdabocconi.it

Riccardo Bonalli
Laboratoire des Signaux et Systèmes, CNRS, CentraleSupélec
Université Paris-Saclay, Gif-sur-Yvette, France
riccardo.bonalli@cnrs.fr

Abstract

We address the problem of safely learning controlled stochastic dynamics from discrete-time trajectory observations, ensuring system trajectories remain within predefined safe regions during both training and deployment. Safety-critical constraints of this kind are crucial in applications such as autonomous robotics, finance, and biomedicine. We introduce a method that ensures safe exploration and efficient estimation of system dynamics by iteratively expanding an initial known safe control set using kernel-based confidence bounds. After training, the learned model enables predictions of the system's dynamics and permits safety verification of any given control. Our approach requires only mild smoothness assumptions and access to an initial safe control set, enabling broad applicability to complex real-world systems. We provide theoretical guarantees for safety and derive adaptive learning rates that improve with increasing Sobolev regularity of the true dynamics. Experimental evaluations demonstrate the practical effectiveness of our method in terms of safety, estimation accuracy, and computational efficiency.

1 Introduction

We consider the problem of safely learning the dynamics of controlled continuous-time stochastic systems from discrete-time observations of trajectory data. This setting is common in applications such as robotics, finance, and healthcare, where system dynamics are only partially known and must be estimated from data. A key challenge in these applications is ensuring safety during both the learning phase and subsequent deployment [1; 2]. As an example, consider an autonomous robot navigating a partially known and turbulent environment, as illustrated in Figure 1. While the deterministic part of the dynamics may be approximately modeled using prior knowledge, the stochastic disturbances (represented by the brown region in Figure 1) due to wind or sensor noise are often unknown and must be learned. Collecting data through naive exploration can result in unsafe trajectories, potentially causing damage to the system or its environment. A second example arises in financial portfolio management, where the drift component of asset prices may be known from historical data, but market volatility remains uncertain. Safety here may correspond to the requirement

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

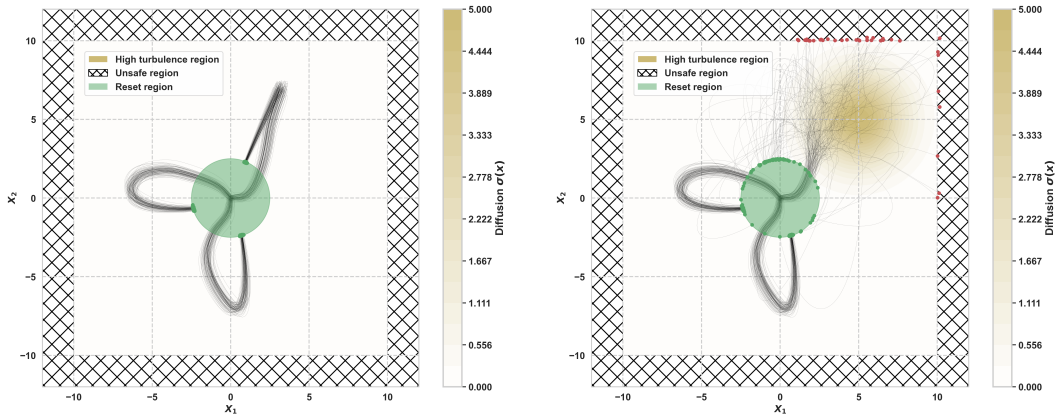


Figure 1: Illustration of a complex, smooth dynamical system under deterministic conditions (left) and stochastic conditions with unknown disturbances (right). Shown are 100 simulated trajectories under three different controls. Ignoring stochastic disturbances (e.g., wind turbulence) can lead to unsafe trajectories (right), emphasizing the necessity of safe estimation methods that explicitly account for uncertainty.

that the portfolio value stays above a critical threshold with high probability, simulating portfolio loss aversion in risk-sensitive financial decision-making. These examples highlight a common need: learning stochastic dynamics of a system from data, while ensuring safety throughout the process. This requires ensuring that all executed trajectories remain within a predefined safe region with high probability. In addition, at deployment time, the learned model should enable prediction of whether a proposed control input satisfies the safety requirements, including those not encountered during training [3; 4].

Outline of contributions. The contributions of this work are as follows.

- **Safe learning method.** We derive a method that safely learns controlled stochastic dynamics, where safety is defined as the requirement that system trajectories remain within a designated set of safe states with high probability. Our approach incrementally expands the known safe control set by selecting novel controls to evaluate, collecting corresponding trajectory data, and refining three models: a dynamics model for predicting state evolution, a safety model that estimates the probability of remaining within the safe region, and a reset model that captures the probability of returning the system to its initial state distribution, enabling repeated safe exploration under uncertainty. Alongside these models, we refine uncertainty estimates using kernel-based confidence bounds. After training, these models enable prediction of dynamics, safety, and reset feasibility for any given control, including those not seen during training.
- **Provably safe exploration and adaptive estimation rates.** We prove that the proposed method guarantees safety and derive learning rates for system estimation, with rates that are adaptive to the Sobolev regularity of the underlying dynamics. Crucially, our approach requires only smooth dynamics (with respect to time, state, and control variables) and an initial non-empty set of known safe controls. These mild assumptions make the method applicable to a broad range of complex real-world systems subject to stochastic disturbances, found in diverse areas including robotics, fluid flow control, and chemical reaction control [5; 6; 7; 8; 9; 10].
- **Experimental validation.** We empirically demonstrate the performance of the approach in terms of safety, estimation accuracy, and computational efficiency. Specifically, we evaluate it on a benchmark two-dimensional stochastic dynamical system evolving in a bounded, safety-critical environment under stochastic perturbations (see Figure 1). An open-source Python implementation is provided (available at `github.com/lmotte/dynamics-safe-learn`).

2 Background

We formalize the problem of safely learning controlled stochastic dynamical systems as follows.

Controlled SDE. Let X be a dynamical system governed by a non-linear, n-dimensional controlled SDE [11; 12],

$$dX(t) = b(X(t), u(t, X(t)))dt + a(X(t), u(t, X(t)))dW(t), \quad u \in \mathcal{H}, \quad X(0) \sim p_0. \tag{1}$$

Here, $T_{\max} > 0$ is the fixed time horizon, (b, a) are functions mapping $\mathbb{R}^n \times \mathbb{R}^d$ to $\mathbb{R}^n \times \mathbb{R}^{n \times n}$, $W(t)$ is an n-dimensional standard Brownian motion, p_0 is an initial probability density over $\mathbb{R}^n$, and $\mathcal{H} \subseteq \mathcal{F}([0, T_{\max}] \times \mathbb{R}^n, \mathbb{R}^d)$ is a finite-dimensional set of admissible controls, where $\mathcal{F}([0, T_{\max}] \times \mathbb{R}^n, \mathbb{R}^d)$ denotes the space of measurable functions from $[0, T_{\max}] \times \mathbb{R}^n$ to $\mathbb{R}^d$.

Example 1 (Second-order dynamical system)**.** *The system*

$$dX(t) \;=\; V(t)\,dt, \quad dV(t) \;=\; u\big(t, X(t)\big)\,dt \;+\; a\big(X(t)\big)\,dW(t),$$

captures second-order dynamics under control u and state-dependent diffusion $a\big(X(t)\big)$. Real-world examples include drones navigating turbulent environments (where u represents thrust or steering, and $a(\cdot)$ models wind turbulence), fluid-dynamical systems [13], and molecular dynamics (where u captures external forces such as optical traps, and $a(\cdot)$ reflects spatially varying thermal fluctuations [14; 15; 16]).

Safety-critical environments. Let $g : \mathbb{R}^n \to \mathbb{R}$ be a function that partitions the state space into safe and unsafe regions. The *safe region* is given by $\{x \in \mathbb{R}^n : g(x) \geq 0\}$, while the *unsafe region* is given by $\{x \in \mathbb{R}^n : g(x) < 0\}$.

Safe control. Let X_u denote the solution to Eq. (1) under the control $u \in \mathcal{H}$ (well-defined by existence and uniqueness; see Sec.3 in [17]). Let $u_\theta : [0, T_{\max}] \times \mathbb{R}^n \to \mathbb{R}^d$ denote a family of controls parameterized by $\theta \in D \subset \mathbb{R}^m$, where D is a compact subset of $\mathbb{R}^m$. Such a finite-dimensional parameterization is a mild assumption that is widely prevalent in many real-world applications, including robotics [18], process control [19], and financial engineering [20].

We define the safety level of the control u_θ *at* time $t \in [0, T_{\max}]$ as

$$s(\theta, t) \triangleq \mathbb{P}\left(g(X_{u_\theta}(t)) \geq 0\right). \tag{2}$$

We define the safety level *up to* time $T \in [0, T_{\max}]$ of the control u_θ as

$$s^\infty(\theta, T) \triangleq \inf_{t \in [0,T]} s(\theta, t). \tag{3}$$

Learning problem. We formulate the problem of safely learning controlled stochastic dynamical systems as the estimation of the probability density map

$$p : (\theta, t, x) \in D \times [0, T_{\max}] \times \mathbb{R}^n \mapsto p_\theta(t, x), \tag{4}$$

where $p_\theta(t, x)$ is the density of the state $X_{u_\theta}(t)$ under control u_θ, well-defined by standard existence and uniqueness results (see Sec.3 in [17]). To estimate this density, we collect a dataset of trajectories

$$(\theta_k, X_{u_{\theta_k}}(w_i^k, t_l))_{k \in \{1,\ldots,K\},\, i \in \{1,\ldots,Q\},\, l \in \{1,\ldots,M_k\}}, \tag{5}$$

where each w_i^k denotes an independent Brownian motion sample path driving the stochastic trajectory, and M_k denotes the number of time steps in trajectory k. All controls u_{θ_k} are required to be safe. Specifically, the minimal probability of remaining within the safe region up to the time horizons $(T_k)_{k=1}^K = (t_{M_k})_{k=1}^K$ is constrained by

$$s^\infty(\theta_k, T_k) \geq 1 - \varepsilon, \quad \text{for each } k \in [\![1, K]\!]. \tag{6}$$

This problem poses a fundamental challenge due to the coupling between learning and safety: accurately estimating the density p_θ requires data, but collecting data must respect safety constraints defined by s^∞, which themselves depend on the very dynamics encoded in p_θ.

2.1 Related work

Safe learning in control systems under uncertainty is a central topic in reinforcement learning, with methods built on assumptions such as known dynamics [21], controllability [22; 23; 24], or recovery policies [25; 26].

Much of the literature focuses on discrete-time or discrete-state systems modeled as Markov Decision Processes (MDPs), where risk-sensitive and safe exploration techniques have been developed [27; 28; 26; 29]. Among these, Safe Bayesian Optimization (BO) methods stand out for providing some of the strongest high-confidence safety guarantees during exploration [30; 31; 32; 33], particularly in MDP settings with known dynamics or access to safety level evaluations [22; 34]. In continuous domains, early work focused on designing control policies that avoid unsafe regions [35; 36], while more recent approaches incorporate offline learning and online adaptation for nonlinear systems [37]. Stability-based guarantees via Lyapunov theory offer formal certification but often require full dynamics knowledge [18; 38]. Safe BO has also been adapted to continuous settings [39; 40], under assumptions such as access to dynamics, safety oracles, or specific control-theoretic properties. Joint estimation of both dynamics and safety remains comparatively underexplored, particularly in continuous-time settings [41; 42; 43; 44; 45; 46; 47; 48; 49].

In contrast to much of the literature, we assume no prior model of the system dynamics or safety function. Instead, we jointly explore and learn both the stochastic dynamics and the safety probabilities from trajectory data. Our method applies to broad classes of continuous-time, continuous-state stochastic systems, and provides provable guarantees on both safety and estimation accuracy. To the best of our knowledge, no prior work provides joint safe exploration and density estimation guarantees in this setting.

3 Assumptions

As formalized by the No-Free-Lunch Theorem [50], learning is only possible under prior assumptions. We now state and discuss the key assumptions used in this work.

Assumption (A1) (Initial safe controls)**.** *For $\varepsilon \in [0,1]$, a non-empty set $S_0 \subset D \times [0, T_{\max}]$ is provided such that*

$$s(\theta, t) \geq 1 - \varepsilon \quad \textit{for all} \quad (\theta, t) \in S_0. \tag{7}$$

This assumption ensures that at least one control is known to be safe at the outset, allowing safe exploration to begin. In fact, S_0 may be as small as a singleton; only one known safe control is required. Without such a point, safe learning cannot be initiated. We express the assumption in set form to allow for larger safe sets, which can accelerate exploration while preserving guarantees. This is a standard assumption in the literature of safe UCB methods [30; 22; 18], and is realistic in many applications including robotics [39] and safety-critical process control [51], where systems naturally start in safe conditions, e.g., $S_0 = \{(0, \theta) : \theta \in D\}$.

Resetting control. Let $h : \mathbb{R}^n \to \mathbb{R}$ define a region in the state space from which resets are feasible. Specifically, if $h(X(t)) \geq 0$, then it is feasible to reset the system to the initial distribution p_0. Formally, this means there exists a mechanism (or control) that reinitializes the system from the current state $X(t)$ to a new state independently sampled from p_0. We define the reset level at time $t \in [0, T_{\max}]$ for a given control u_θ as

$$r(\theta, t) \triangleq \mathbb{P}\left(h(X_{u_\theta}(t)) \geq 0\right). \tag{8}$$

The function h delimits a region of the state space from which resets are feasible. Larger reset regions correspond to greater operational flexibility. In simulated environments, where resets are effectively cost-free, the reset region can cover the entire state space $\mathbb{R}^n$. In contrast, real-world systems typically require substantial resources or manual intervention, making resets feasible only in restricted regions (e.g., near the original distribution p_0).

Assumption (A2) (Initial resetting controls)**.** *For $\xi \in [0,1]$, a non-empty set $R_0 \subset D \times [0, T_{\max}]$ is provided such that*

$$r(\theta, t) \geq 1 - \xi \quad \textit{for all} \quad (\theta, t) \in R_0. \tag{9}$$

This assumption guarantees the existence of at least one control capable of returning the system to the reset region with high probability. Assumption (A2) enables the generation of independent sample paths starting from the same initial conditions, which is crucial for evaluating variance and managing uncertainty during safe exploration. As with Assumption (A1), one known reset point is sufficient, though larger reset sets accelerate learning. A simple case is $R_0 = \{(0, \theta) : \theta \in D\}$, corresponding to systems that can always be reset from the initial condition. In practice, reset feasibility depends on system constraints: for instance, batch chemical reactors can often be reset only in early phases, before irreversible reactions occur [19]. In contrast, many autonomous systems like drones or driving robots can usually be reset, at least during training.

Assumption (A3) (Smoothness of system dynamics)**.** *The map p lies in the Sobolev space $H^\nu(\mathbb{R}^{n+m+1})$ with $\nu > \frac{1}{2}\max(n, m+1)$, where n and m denote the state and control parameter dimensions, respectively. Moreover,* $\sup_{x\in\mathbb{R}^n} \left\|p(\cdot,\cdot,x)\right\|_{H^\nu(\mathbb{R}^{m+1})} < +\infty$, $\sup_{(\theta,t)\in D\times[0,T_{\max}]} \|p(\theta,t,\cdot)\|_{H^\nu(\mathbb{R}^n)} < \infty$.

This smoothness assumption ensures that the system dynamics are sufficiently regular for our purposes. It is standard in statistical learning theory and underpins our convergence guarantees [52]. Sobolev regularity of the drift and diffusion terms in the underlying stochastic differential equation is expected to imply Sobolev regularity of the resulting state densities under standard conditions. This follows from classical results in parabolic PDE theory, where solutions typically gain regularity relative to the coefficients, roughly two derivatives in space and one in time. A formal analysis of this connection is beyond the scope of the present work and is left for future investigation (see Bonalli and Rudi [17] for related results).

4 Proposed method

We propose a method for safely exploring and learning system dynamics over a parameterized control space $\mathcal{H} = \{u_\theta \mid \theta \in D \subset \mathbb{R}^m\}$. Following the safe UCB framework [30; 31; 32], our goal is to select controls that reduce model uncertainty while ensuring, with high probability, that trajectories (i) remain within the safe region and (ii) end in the reset region. We jointly learn three models: a dynamics model (state densities), a safety model (safety probabilities), and a reset model (reset probabilities), each equipped with confidence bounds from a shared kernel. This enables active exploration under high-probability constraints. After training, the learned models support inference on unseen inputs and yield a certified control set that can be deployed with safety guarantees.

The known safe-resettable set is expanded iteratively by alternating between system estimation (Section 4.2) and safe sampling (Section 4.3), leveraging prior knowledge of the initial safe and reset sets as well as the regularity of the dynamics $(\theta, t, x) \mapsto p_\theta(t, x)$.

A step-by-step breakdown of the overall method is provided in Appendix B, with algorithm tables for each module and their computational complexities.

4.1 Initialization

Let $N \in \mathbb{N}$ denote the current iteration. We initialize at $N = 0$ using the known safe set $S_0 \subset D \times [0, T_{\max}]$ and reset set $R_0 \subset D \times [0, T_{\max}]$. We define the initial safe-resettable set

$$\Gamma_0 \triangleq \left\{(\theta, t, T) \in D \times [0, T_{\max}]^2 \,\middle|\, t \le T,\ (\theta, t') \in S_0 \text{ for all } t' \in [0, T],\ (\theta, T) \in R_0\right\}.$$

We select $(\theta_0, t_0, T_0) \in \Gamma_0$, ensuring that the control is known to be safe over $[0, T_0]$ and ends in the reset region.

4.2 System estimation

In this step, we update the dynamics, safety, and reset models based on the observed trajectories, and compute predictive uncertainty for each.

Estimation at (θ_N, t_N). At iteration N, the control u_{θ_N} is evaluated using Q stochastic trajectories. Here, N indexes the iteration, and i indexes the i-th trajectory simulated under that control, each corresponding to an independent sample w_i^N of the Brownian motion driving the system. We collect

the samples $(X_{u_{\theta_N}}(w_i^N, t_N))_{i=1}^Q$ and estimate the state density at (θ_N, t_N) using a kernel density estimator:

$$\hat{p}_{\theta_N,t_N}(x) \triangleq \frac{1}{Q}\sum_{i=1}^{Q} \rho_R(x - X_{u_{\theta_N}}(w_i^N, t_N)), \tag{10}$$

where $\rho_R(x) \triangleq R^{n/2}\|x\|^{-n/2}B_{n/2}(2\pi R\|x\|)$, $R > 0$, and $B_{n/2}$ is the Bessel J function of order $n/2$ (See Bonalli and Rudi [17]).

We then compute estimates of the safety and reset probabilities

$$\hat{s}_{\theta_N,t_N} \triangleq \int_{\{x\in\mathbb{R}^n : g(x)\geq 0\}} \hat{p}_{\theta_N}(t_N, x)\, dx, \quad \hat{r}_{\theta_N,t_N} \triangleq \int_{\{x\in\mathbb{R}^n : h(x)\geq 0\}} \hat{p}_{\theta_N}(t_N, x)\, dx. \tag{11}$$

Let the collection of values at all observed points $((\theta_i, t_i))_{i=1}^N$ be

$$\hat{P}(\cdot) \triangleq (\hat{p}_{\theta_i,t_i}(\cdot))_{i=1}^N, \quad \hat{S} \triangleq (\hat{s}_{\theta_i,t_i})_{i=1}^N, \quad \hat{R} \triangleq (\hat{r}_{\theta_i,t_i})_{i=1}^N.$$

Model update. We fit kernel ridge regressors for the density, safety, and reset functions using a Matérn kernel k (with Sobolev smoothness ν) and regularization $\lambda > 0$

$$\hat{p}_\theta(t, x) \triangleq \hat{P}(x)(K + N\lambda I)^{-1}k(\theta, t), \qquad \text{(System dynamics)} \tag{12}$$

$$\hat{s}_N(\theta, t) \triangleq \hat{S}(K + N\lambda I)^{-1}k(\theta, t), \qquad \text{(Safety function)} \tag{13}$$

$$\hat{r}_N(\theta, t) \triangleq \hat{R}(K + N\lambda I)^{-1}k(\theta, t), \qquad \text{(Reset function)} \tag{14}$$

where $k(\theta, t) \triangleq (k((\theta, t), (\theta_i, t_i)))_{i=1}^N$, $K \triangleq (k((\theta_i, t_i), (\theta_j, t_j)))_{i,j=1}^N$, and λ is a regularization term. Although training data consist of discrete-time observations, learned regression models are defined over continuous time, a distinction seldom addressed in the literature.

The predictive uncertainty at (θ, t) is given by

$$\sigma_N^2(\theta, t) \triangleq k((\theta, t), (\theta, t)) - k(\theta, t)^*(K + N\lambda I)^{-1}k(\theta, t). \tag{15}$$

4.3 Safe sampling

We now select a new point to sample by maximizing uncertainty over the safe-resettable region.

Feasibility criteria. A point (θ, t, T) is feasible if (i) $t \leq T$, (ii) the system remains safe up to time T, i.e., $s^\infty(\theta, T) \geq 1 - \varepsilon$, and (iii) the trajectory ends in the reset region with high probability, i.e., $r(\theta, T) \geq 1 - \xi$.

We implement these constraints via lower confidence bounds (LCBs)

$$\mathrm{LCB}_{\mathrm{N}}^{\mathrm{s}}(\theta, T) \triangleq \inf_{t\in[0,T]} \big(\hat{s}_N(\theta, t) - \beta_N^s \sigma_N(\theta, t)\big), \tag{16}$$

$$\mathrm{LCB}_{\mathrm{N}}^{\mathrm{r}}(\theta, T) \triangleq \hat{r}_N(\theta, T) - \beta_N^r \sigma_N(\theta, T), \tag{17}$$

where $\beta_N^s, \beta_N^r > 0$ are confidence parameters set from known upper bounds on the RKHS norms of the safety and reset functions (see Remark 1).

We then define the safe-resettable feasible set

$$\Gamma_N = \Gamma_0 \cup \Big\{(\theta, t, T) \in D \times [0, T_{\max}]^2 \mid t \leq T,\ \mathrm{LCB}_{\mathrm{N}}^{\mathrm{s}}(\theta, T) \geq 1 - \varepsilon,\ \mathrm{LCB}_{\mathrm{N}}^{\mathrm{r}}(\theta, T) \geq 1 - \xi\Big\}.$$

Sampling rule. We choose the next $(\theta_{N+1}, t_{N+1}, T_{N+1})$ by maximizing uncertainty over the feasible set

$$(\theta_{N+1}, t_{N+1}, T_{N+1}) = \operatorname*{arg\,max}_{(\theta,t,T)\in\Gamma_N} \sigma_N(\theta, t). \tag{18}$$

Optimization is performed using discretization or gradient-based methods. Several computational techniques for efficient optimization are presented in the Appendix (see Algorithm 4).

Stopping rule. We stop the exploration once the maximum uncertainty over the feasible set falls below a threshold $\eta > 0$

$$\max_{(\theta,t,T)\in\Gamma_N} \sigma_N(\theta,t) < \eta, \tag{19}$$

ensuring that exploration concludes once the models reaches the desired level of accuracy.

Remark 1. *The derivation in Appendix A.7 shows that the confidence parameters depend on upper bounds of the RKHS norms of the safety and reset functions. These bounds may be available from prior knowledge of the system's regularity. We view this prior knowledge as a reasonable minimal assumption for guaranteeing safe learning under unknown dynamics. When unavailable, the bounds can be conservatively overestimated, ensuring safety but potentially leading to slower exploration. Developing adaptive strategies to estimate these quantities without prior knowledge is a promising direction for future work, for instance through online adaptation via the doubling trick [53].*

5 Safety and estimation guarantees for Sobolev dynamics

Safety and exploration guarantees for safe kernelized UCB methods have been developed in prior work [30; 31; 32], grounded in kernelized bandit theory [54; 55; 56], which in turn builds on linear bandit results [57; 58]. Building on this foundation, we establish novel theoretical guarantees for safe exploration and dynamics estimation under Sobolev regularity. Complete proofs are deferred to Appendix A.

Theorem 5.1 (Safely learning controlled Sobolev dynamics)**.** *Let $\eta > 0$, and assume Assumptions (A1)–(A3) hold. Set $R = Q^{1/(n+2\nu)}$ and $\lambda = N^{-1}$. Then there exist constants $c_1, \dots, c_5 > 0$, independent of N, Q, δ, η, such that if*

$$c_1 \log(4N/\delta)^{1/2} Q^{\frac{n-2\nu}{2n+4\nu}} \leq N^{-1/2},^{1}$$

then the stopping condition $\max_{(\theta,t,T)\in\Gamma_N} \sigma_N(\theta,t) < \eta$ is satisfied after at most $N \leq c_2 \eta^{-2/(1-\alpha)}$ iterations for any $\alpha > (m+1)/(m+1+2\nu)$. Moreover:

- ***(Safety):** All selected triples (θ_i, t_i, T_i) satisfy $s^\infty(\theta_i, T_i) \geq 1-\varepsilon$ and $r(\theta_i, T_i) \geq 1-\xi$, providing safety guarantees during training. Moreover, the final set Γ_N includes only controls meeting these thresholds and can thus serve as a certified safe set for deployment.*
- ***(Estimation guarantees):** For all $(\theta, t, T) \in \Gamma_N$,*

$$\|\hat{p}_\theta(t,\cdot) - p_\theta(t,\cdot)\|_\infty \leq c_3\eta, \quad |\hat{s}_N(\theta,t) - s(\theta,t)| \leq c_4\eta, \quad |\hat{r}_N(\theta,t) - r(\theta,t)| \leq c_5\eta.$$

This result ensures that our method both respects safety constraints and achieves convergence rates adaptive to the system's Sobolev regularity. The condition on Q provides a lower bound on the number of trajectory samples Q required per control to guarantee the prescribed confidence level. Up to logarithmic factors, it requires

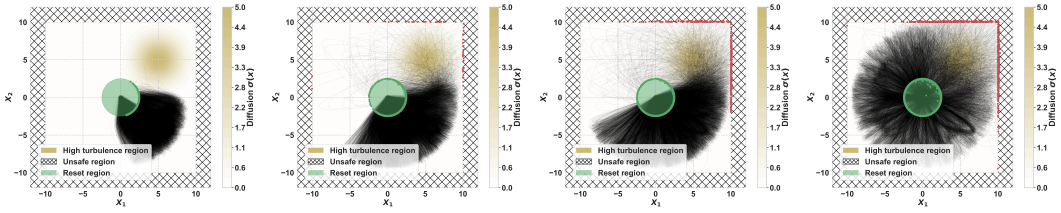
$$Q \gtrsim N^{\frac{2\nu+n}{2\nu-n}},$$

where n is the state dimension and ν the Sobolev regularity. For instance, when the system is sufficiently regular with $\nu \geq n+m+1$, the algorithm terminates in at most $N = \mathcal{O}(\eta^{-3})$ iterations, assuming $Q \gtrsim N^3$. Although we do not analyze the size of Γ_N, its structure can be inferred from the available uncertainty estimates; a formal characterization of Γ_N is left to future work.

6 Numerical experiments

We evaluate our method on a representative smooth nonlinear stochastic system. Specifically, the experiments aim to assess the following: (i) satisfaction of safety and reset constraints, (ii) efficiency of exploration under different safety thresholds, (iii) prediction accuracy for dynamics, safety, and reset maps, (iv) computational cost and scalability.

[1] All exponents in n are to be read as $n+\varepsilon$, with $\varepsilon > 0$ arbitrarily small.



$\varepsilon = \xi = 0.1$ $\varepsilon = \xi = 0.3$ $\varepsilon = \xi = 0.5$ $\varepsilon = \xi = \infty$

Figure 2: One trajectory per selected control after 1000 iterations for various thresholds.

System and environment. We consider a 2D second-order dynamical system whose acceleration is directly controlled by the input. The system evolves according to the controlled SDE

$$\begin{cases} dX(t) = V(t)dt, \\ dV(t) = u(t, X(t), V(t))dt + a(X(t))dW_t. \end{cases}$$

where $X(t) \in \mathbb{R}^2$, $V(t) \in \mathbb{R}^2$ denote position and velocity, u is the control function, and W_t is a Brownian motion. The noise amplitude $a(X)$ is spatially dependent:

$$a(X) = A \exp\left(-\frac{\|X - X_c\|^2}{2\sigma^2}\right),$$

with $X_c = (5, 5)$, $\sigma = 2$, and $A = 5$. The initial state follows $\mathcal{N}(0, \sigma_0 I_{\mathbb{R}^2})$ with $\sigma_0 = 0.1$, and the maximal time horizon is $T_{\max} = 20$. The system evolves within the bounded safe region $(-10, 10)^2$, and each trajectory must end in the reset region defined as a disk of radius 2.5 centered at the origin. Such models arise in robotics and autonomous navigation, involving trajectory control with localized disturbances (e.g., slippery or uneven terrain). Figure 1 illustrates the effect of such state-dependent noise through 100 trajectories generated under different controls.

Control space. Controls are parameterized as sequences of m fixed accelerations of magnitude v, applied in directions $(\theta_1, \ldots, \theta_m)$. During the exploration phase $(0 \leq t \leq T_{\text{explo}})$, each direction θ_i is applied over intervals of equal length, yielding

$$u(t, X, V) = v(\cos(\theta_i), \sin(\theta_i)) - V,$$

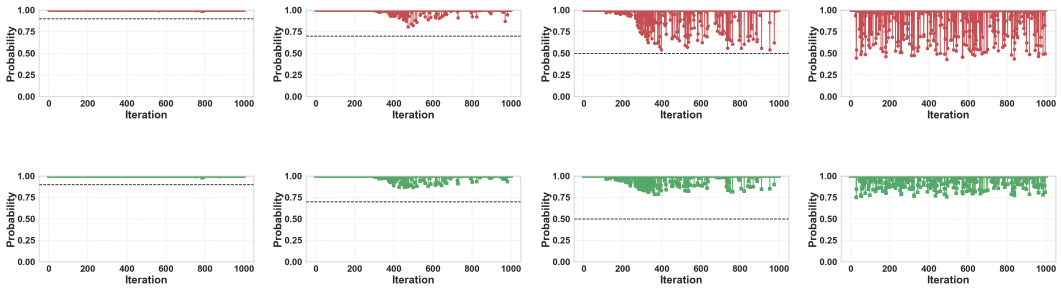
with damping term $-V$ ensuring velocity convergence. For $(t > T_{\text{explo}})$, a feedback controller steers the system toward μ_0:

$$u(t, X, V) = \kappa \times \left(v \frac{\mu_0 - X}{\|\mu_0 - X\|} - V\right),$$

with damping factor $\kappa > 0$. Controls are clipped to keep the system within the safe region. We set $v = 2.0$, $\kappa = 0.5$, $m = 2$, $T_{\text{explo}} = 6$, and $n_{\text{steps}} = 500$.

Method's hyperparameters. Our method depends on several hyperparameters that govern safety thresholds (ε, ξ), confidence levels (β_s, β_r), kernel smoothness (λ, γ), and bandwidth R, with distinct values for estimating dynamics and constraints. We test $(\varepsilon, \xi) \in \{0.1, 0.3, 0.5, \infty\}$, with 1000 iterations and initial safe control $(-\pi/3, \pi/3)$. Candidate selection for uncertainty maximization is restricted to a local subset for efficiency (Appendix B, Algorithm 4). A detailed discussion of each hyperparameter's role, tuning procedure, and practical heuristics is provided in Appendix B.3.

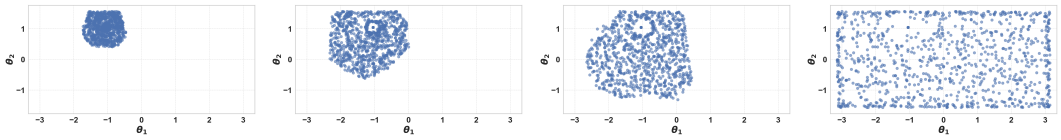
Safe exploration. Figure 2 displays one trajectory per selected control after 1000 iterations, under various threshold settings $(\varepsilon = \xi \in \{0.1, 0.3, 0.5, +\infty\})$. In Figure 3, the top row displays the learned safety level maps while the bottom row shows the corresponding reset probability maps for various threshold pairs $\varepsilon = \xi$, with values increasing from left to right in $\{0.1, 0.3, 0.5, +\infty\}$. Our approach only accepts candidate controls whose predicted safety and reset probabilities (estimated via 200-path Monte Carlo simulations) exceed the predefined thresholds. By filtering only controls meeting the safety criteria, exploration is confined to a safe region with a chosen probability of staying safe. Overall, these visualizations highlight how increasing the threshold values influences control selection, providing insights into the trade-off between exploration and safety.



$\varepsilon = \xi = 0.1$ $\varepsilon = \xi = 0.3$ $\varepsilon = \xi = 0.5$ $\varepsilon = \xi = \infty$

Figure 3: Safety (top row) and reset (bottom row) probabilities over iterations for various thresholds.

Exploration rate and coverage. In Figure 4, we plot the selected controls after 1000 iterations for various threshold pairs $\varepsilon = \xi$, with values increasing from left to right in $\{0.1,\ 0.3,\ 0.5, +\infty\}$. Figures 2 and 4 clearly illustrate the exploration-safety trade-off. Relaxing thresholds leads to broader control coverage and faster information gain, but with decreased safety guarantees. Conversely, strict thresholds restrict exploration, particularly around regions with safety or reset probabilities close to the specified thresholds. This aligns with the intuition supported by our theoretical analysis: sample complexity tends to increase in regions where smaller uncertainty is required to proceed safely. As a result, these regions act as bottlenecks, slowing down the process and potentially stopping exploration within the connected component that satisfies the constraints and includes the initial safe control. Additional results on information gain across iterations are provided in Appendix C.2.



$\varepsilon = \xi = 0.1$ $\varepsilon = \xi = 0.3$ $\varepsilon = \xi = 0.5$ $\varepsilon = \xi = \infty$

Figure 4: Control coverage for various thresholds.

Safety and reset level prediction. In Table 1, we quantify the accuracy of the learned model for various threshold pairs $\varepsilon = \xi$ in $\{0.1,\ 0.3,\ 0.5, +\infty\}$ by evaluating the prediction quality of the safety and reset levels over 1000 predictions. We report the mean squared error (MSE) and the standard deviation, with the ground truth provided by Monte Carlo estimates based on 100 samples (displayed in Figure 5). In Figure 6, we plot the learned safety and reset maps, whose accuracies can be qualitatively assessed by comparing their values with those in Figure 5. As expected, prediction accuracy improves as safety constraints are relaxed, due to the broader coverage of the control space.

Table 1: Safety and reset level prediction error statistics (MSE $\pm$ Std. Dev.)

Model (ε, ξ)	**Safety MSE**	**Reset MSE**
$(0.1, 0.1)$	0.7010 ± 0.3847	0.6919 ± 0.3764
$(0.3, 0.3)$	0.5217 ± 0.4254	0.5197 ± 0.4161
$(0.5, 0.5)$	0.3736 ± 0.4299	0.3701 ± 0.4258
$(+\infty, +\infty)$	0.0023 ± 0.0065	0.0024 ± 0.0062

Dynamics prediction. To verify that our method captures the underlying system dynamics, we compare predicted trajectory densities with ground-truth trajectories under known-safe controls. Qualitative results show close agreement in both mean and variance. Full visualizations and evaluation details are provided in Appendix C.1.

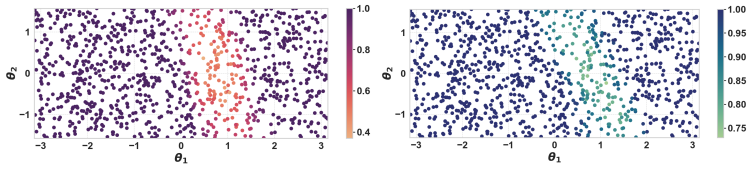
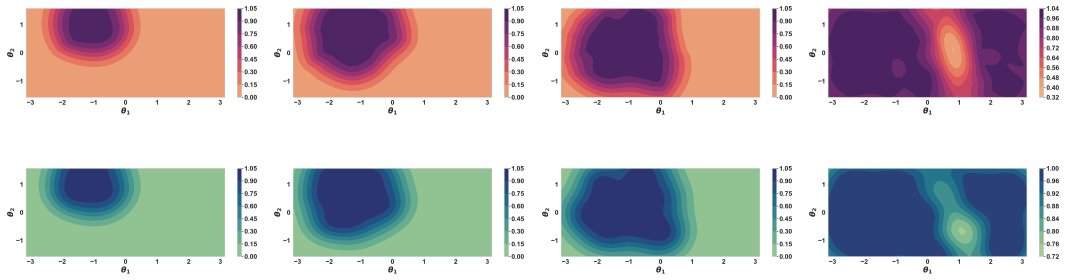


Figure 5: Ground-truth safety (left) and reset (right) probabilities estimated via 100 Monte Carlo samples for 1000 randomly selected controls.

Computational considerations. Our method runs end-to-end in under 32 minutes on standard hardware, covering candidate selection, simulation, evaluation, and model updates. Appendix B.4 provides runtimes, hardware specs, and potential optimizations (e.g., sketching, parallelization), confirming the method's practicality on standard hardware.



$\varepsilon = \xi = 0.1$ $\varepsilon = \xi = 0.3$ $\varepsilon = \xi = 0.5$ $\varepsilon = \xi = \infty$

Figure 6: Learned safety (top row) and reset (bottom row) level maps for various thresholds.

7 Conclusion

We introduced a provably safe and efficient method for learning controlled stochastic dynamics from trajectory data. By leveraging kernel-based confidence bounds and smoothness assumptions, our method incrementally expands an initial safe control set, ensuring that all trajectories remain within predefined safety regions throughout the learning process. Theoretical guarantees were established for both safety and estimation accuracy, with learning rates that adapt to the Sobolev regularity of the true dynamics. Numerical experiments corroborate our theoretical findings regarding safety and estimation accuracy. By tuning the safety (ε) and reset (ξ) thresholds, users can explicitly control the trade-off between conservative safety satisfaction and exploratory behavior. While our experimental validation focuses on a low-dimensional setting, the theoretical results scale with dimension: the convergence rates for the proposed estimators decrease polynomially with dimension and can mitigate the curse of dimensionality under sufficient smoothness. This makes the method applicable to higher-dimensional systems, which we plan to investigate in future work. Further research will include validation on physical systems (e.g., autonomous robots), improved scalability through fast kernel methods (e.g., sketching or incremental updates), comparisons with safe RL baselines, systematic analyses of kernel and threshold selection, and extensions to handle abrupt dynamics and non-diffusive disturbances such as jump processes arising in pedestrian–vehicle interactions and hybrid systems. These developments will further support applications in safety-critical control and decision-making under uncertainty.

Acknowledgements

The Agence Nationale de la Recherche (grant ANR-22-CE48-0006, PI: R.B.) provided funds to assist the authors with their research. A.R. acknowledges support from the European Research Council (grant REAL 947908).

References

[1] R. Bonalli, T. Lew, and M. Pavone. Sequential Convex Programming for Non-linear Stochastic Optimal Control. *ESAIM: Control, Optimisation and Calculus of Variations*, 28:64, 2022.

[2] T. Lew, R. Bonalli, and M. Pavone. Sample Average Approximation for Stochastic Programming with Equality Constraints. *SIAM Journal on Optimization*, 34:3506–3533, 2024.

[3] Kim Peter Wabersich and Melanie N Zeilinger. A predictive safety filter for learning-based control of constrained nonlinear dynamical systems. *Automatica*, 129:109597, 2021.

[4] Lars Lindemann, Matthew Cleaveland, Gihyun Shim, and George J Pappas. Safe planning in dynamic environments using conformal prediction. *IEEE Robotics and Automation Letters*, 8(8):5116–5123, 2023.

[5] Philippe Martin and Erwan Salaün. The true role of accelerometer feedback in quadrotor control. In *2010 IEEE international conference on robotics and automation*, pages 1623–1629. IEEE, 2010.

[6] Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob McGrew, Bowen Baker, Glenn Powell, Jonas Schneider, Josh Tobin, Maciek Chociej, Peter Welinder, et al. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464*, 2018.

[7] Jeremy Morton, Antony Jameson, Mykel J Kochenderfer, and Freddie Witherden. Deep dynamical modeling and control of unsteady fluid flows. *Advances in Neural Information Processing Systems*, 31, 2018.

[8] Jonas Peters, Stefan Bauer, and Niklas Pfister. *Causal Models for Dynamical Systems*, page 671–690. Association for Computing Machinery, New York, NY, USA, 1 edition, 2022. ISBN 9781450395861. URL `https://doi.org/10.1145/3501714.3501752`.

[9] Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation using low-cost whole-body teleoperation. In *8th Annual Conference on Robot Learning*, 2024.

[10] Long Wei, Peiyan Hu, Ruiqi Feng, Haodong Feng, Yixuan Du, Tao Zhang, Rui Wang, Yue Wang, Zhi-Ming Ma, and Tailin Wu. Diffphycon: A generative approach to control complex physical systems. *arXiv preprint arXiv:2407.06494*, 2024.

[11] R. Bonalli and B. Bonnet. First-order Pontryagin Maximum Principle for Risk-averse Stochastic Optimal Control Problems. *SIAM Journal on Control and Optimization*, 61:1881–1909, 2023.

[12] T. Lew, R. Bonalli, and M. Pavone. Risk-averse Trajectory Optimization via Sample Average Approximation. *IEEE Robotics and Automation Letters*, 9:1500–1507, 2023.

[13] George Keith Batchelor. *An introduction to fluid dynamics*. Cambridge university press, 2000.

[14] Giorgio Volpe and Giovanni Volpe. Simulation of a brownian particle in an optical trap. *American Journal of Physics*, 81(3):224–230, 2013.

[15] Onofrio M Marago, Philip H Jones, Pietro G Gucciardi, Giovanni Volpe, and Andrea C Ferrari. Optical trapping and manipulation of nanostructures. *Nature nanotechnology*, 8(11):807–819, 2013.

[16] Giuseppe Pesce, Philip H Jones, Onofrio M Maragò, and Giovanni Volpe. Optical tweezers: theory and practice. *The European Physical Journal Plus*, 135:1–38, 2020.

[17] Riccardo Bonalli and Alessandro Rudi. Non-parametric learning of stochastic differential equations with non-asymptotic fast rates of convergence. *Foundations of Computational Mathematics*, pages 1–56, 2025.

[18] Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. *Advances in neural information processing systems*, 30, 2017.

[19] Dale E Seborg, Thomas F Edgar, Duncan A Mellichamp, and Francis J Doyle III. *Process dynamics and control*. John Wiley & Sons, 2016.

[20] Robert C Merton. Optimum consumption and portfolio rules in a continuous-time model. In *Stochastic optimization models in finance*, pages 621–661. Elsevier, 1975.

[21] Lukas Hewing, Kim P Wabersich, Marcel Menner, and Melanie N Zeilinger. Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(1):269–296, 2020.

[22] Matteo Turchetta, Felix Berkenkamp, and Andreas Krause. Safe exploration in finite markov decision processes with gaussian processes. *Advances in neural information processing systems*, 29, 2016.

[23] Horia Mania, Michael I Jordan, and Benjamin Recht. Active learning for nonlinear system identification with guarantees. *Journal of Machine Learning Research*, 23(32):1–30, 2022.

[24] Mahmoud Selim, Amr Alanwar, Shreyas Kousik, Grace Gao, Marco Pavone, and Karl H Johansson. Safe reinforcement learning using black-box reachability analysis. *IEEE Robotics and Automation Letters*, 7(4): 10665–10672, 2022.

[25] Alexander Hans, Daniel Schneegaß, Anton Maximilian Schäfer, and Steffen Udluft. Safe exploration for reinforcement learning. In *ESANN*, pages 143–148, 2008.

[26] Teodor Mihai Moldovan and Pieter Abbeel. Safe exploration in markov decision processes. *arXiv preprint arXiv:1205.4810*, 2012.

[27] Stefano P Coraluppi and Steven I Marcus. Risk-sensitive and minimax control of discrete-time, finite-state markov decision processes. *Automatica*, 35(2):301–309, 1999.

[28] Peter Geibel and Fritz Wysotzki. Risk-sensitive reinforcement learning applied to control under constraints. *Journal of Artificial Intelligence Research*, 24:81–108, 2005.

[29] Javier Garcia and Fernando Fernández. Safe exploration of state and action spaces in reinforcement learning. *Journal of Artificial Intelligence Research*, 45:515–564, 2012.

[30] Yanan Sui, Alkis Gotovos, Joel Burdick, and Andreas Krause. Safe exploration for optimization with gaussian processes. In *International conference on machine learning*, pages 997–1005. PMLR, 2015.

[31] Yanan Sui, Vincent Zhuang, Joel Burdick, and Yisong Yue. Stagewise safe bayesian optimization with gaussian processes. In *International conference on machine learning*, pages 4781–4789. PMLR, 2018.

[32] Alessandro Bottero, Carlos Luis, Julia Vinogradska, Felix Berkenkamp, and Jan R Peters. Information-theoretic safe exploration with gaussian processes. *Advances in Neural Information Processing Systems*, 35:30707–30719, 2022.

[33] Jialin Li, Marta Zagorowska, Giulia De Pasquale, Alisa Rupenyan, and John Lygeros. Safe time-varying optimization based on gaussian processes with spatio-temporal kernel. *Advances in Neural Information Processing Systems*, 37:95326–95355, 2024.

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

[35] Anayo K Akametalu, Jaime F Fisac, Jeremy H Gillula, Shahab Kaynama, Melanie N Zeilinger, and Claire J Tomlin. Reachability-based safe learning with gaussian processes. In *53rd IEEE conference on decision and control*, pages 1424–1431. IEEE, 2014.

[36] Torsten Koller, Felix Berkenkamp, Matteo Turchetta, and Andreas Krause. Learning-based model predictive control for safe exploration. In *2018 IEEE conference on decision and control (CDC)*, pages 6059–6066. IEEE, 2018.

[37] Thomas Lew, Apoorva Sharma, James Harrison, Andrew Bylard, and Marco Pavone. Safe active dynamics learning and control: A sequential exploration–exploitation framework. *IEEE Transactions on Robotics*, 38(5):2888–2907, 2022.

[38] Spencer M Richards, Felix Berkenkamp, and Andreas Krause. The lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems. In *Conference on Robot Learning*, pages 466–476. PMLR, 2018.

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

[40] Manish Prajapat, Johannes Köhler, Matteo Turchetta, Andreas Krause, and Melanie N Zeilinger. Safe guaranteed exploration for non-linear systems. *arXiv preprint arXiv:2402.06562*, 2024.

[41] Amir Ali Ahmadi, Abraar Chaudhry, Vikas Sindhwani, and Stephen Tu. Safely learning dynamical systems from short trajectories. In *Learning for Dynamics and Control*, pages 498–509. PMLR, 2021.

[42] Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerik, Todd Hester, Cosmin Paduraru, and Yuval Tassa. Safe exploration in continuous action spaces. *arXiv preprint arXiv:1801.08757*, 2018.

[43] Kim P Wabersich and Melanie N Zeilinger. Safe exploration of nonlinear dynamical systems: A predictive safety filter for reinforcement learning. *arXiv preprint arXiv:1812.05506*, 2018.

[44] Jaime F Fisac, Anayo K Akametalu, Melanie N Zeilinger, Shahab Kaynama, Jeremy Gillula, and Claire J Tomlin. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Transactions on Automatic Control*, 64(7):2737–2752, 2018.

[45] Kim P Wabersich and Melanie N Zeilinger. Linear model predictive safety certification for learning-based control. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 7130–7135. IEEE, 2018.

[46] Mohammad Javad Khojasteh, Vikas Dhiman, Massimo Franceschetti, and Nikolay Atanasov. Probabilistic safety constraints for learned high relative degree system dynamics. In *Learning for Dynamics and Control*, pages 781–792. PMLR, 2020.

[47] Nolan C Wagener, Byron Boots, and Ching-An Cheng. Safe reinforcement learning using advantage-based intervention. In *International Conference on Machine Learning*, pages 10630–10640. PMLR, 2021.

[48] Akifumi Wachi and Yanan Sui. Safe reinforcement learning in constrained markov decision processes. In *International Conference on Machine Learning*, pages 9797–9806. PMLR, 2020.

[49] Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, and Alois Knoll. A review of safe reinforcement learning: Methods, theory and applications. *arXiv preprint arXiv:2205.10330*, 2022.

[50] Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media, 2013.

[51] S Joe Qin and Thomas A Badgwell. A survey of industrial model predictive control technology. *Control engineering practice*, 11(7):733–764, 2003.

[52] Loucas Pillaud-Vivien, Alessandro Rudi, and Francis Bach. Statistical optimality of stochastic gradient descent on hard learning problems through multiple passes. *Advances in Neural Information Processing Systems*, 31, 2018.

[53] Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

[54] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

[55] Michal Valko, Nathaniel Korda, Rémi Munos, Ilias Flaounas, and Nelo Cristianini. Finite-time analysis of kernelised contextual bandits. *arXiv preprint arXiv:1309.6869*, 2013.

[56] David Janz, David Burt, and Javier González. Bandit optimisation of functions in the matérn kernel rkhs. In *International Conference on Artificial Intelligence and Statistics*, pages 2486–2495. PMLR, 2020.

[57] Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback. In *COLT*, volume 2, page 3, 2008.

[58] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

[59] Matthias W Seeger, Sham M Kakade, and Dean P Foster. Information consistency of nonparametric gaussian process methods. *IEEE Transactions on Information Theory*, 54(5):2376–2382, 2008.

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

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and the introduction clearly state the contributions: a method for safe learning of controlled dynamics, theoretical guarantees under Sobolev regularity, and numerical evaluation of performance. These claims are consistently supported throughout the methodological, theoretical and experimental sections.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations including model assumptions (Sobolev regularity, known initial safe/reset sets), computational cost (see Assumptions, Theory and Experiments sections).

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All theoretical results are stated with assumptions and formally proved in Appendix A. The main theorem clearly specifies conditions on smoothness and sampling.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The main paper and appendix detail the full experimental setup: system dynamics, control parametrization, thresholds, kernel parameters, and iteration budget. All elements needed to reproduce the results are included.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code and data will be made publicly available upon publication, and are shared during the anonymous review phase. Instructions for reproduction are included in the code's documentation.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The system definition, exploration policy, control parameters, thresholds, hyperparameters, and evaluation metrics are all clearly described in Section 6 and Appendix B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Error bars (mean $\pm$ std) are reported for prediction error in Table 1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 6 and Appendix B.4 provide runtimes and hardware details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research adheres to NeurIPS Ethics Guidelines. It does not involve human subjects, personal data, or high-risk deployments.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: The paper addresses safe learning for stochastic control systems, with applications in robotics and autonomous systems. It offers methods to ensure safety during training, which is a positive contribution. We see no direct negative societal impacts from the proposed methodology.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The method does not involve pretrained models or scraped datasets and poses minimal risk of misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All external tools used are standard academic libraries. No external datasets are used.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: No new datasets or models are released. The experiments are synthetic.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The research does not involve crowdsourcing or human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No human subjects are involved in this research.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: No LLMs were used in the design, implementation, or analysis of the core research method.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Proofs

A.1 Notations

We denote by S_+^n the set of positive-definite matrices in $\mathbb{R}^{n\times n}$. The space of all measurable functions mapping a set A to a set B is denoted by $\mathcal{F}(A,B)$. For any vectors u, v, their tensor product is denoted by $u \otimes v$. For any conformable operator A, we define $A_\lambda \triangleq A + \lambda I$, where I is the identity operator. The minimum and maximum between two scalars a, b are denoted as $a \wedge b \triangleq \min(a,b)$ and $a \vee b \triangleq \max(a,b)$. To quantify function smoothness, we use Sobolev spaces. For a domain $\Omega \subset \mathbb{R}^d$, the Sobolev space $H^\nu(\Omega)$ consists of functions whose weak derivatives up to order ν exist and are square-integrable. The Sobolev embedding theorem states that if $\nu > d/2 + k$, then functions in $H^\nu(\Omega)$ are at least $C^k(\Omega)$-smooth. In our work, we consider domains such as $\mathbb{R}^n$ for spatial variables and $D \times [0, T_{\max}]$ for control-time spaces , where $D \subset \mathbb{R}^m$ is a compact set of control parameters. Finally, for the RKHS $\mathcal{G}$ on $D \times [0, T_{\max}]$ with kernel k, and any $p : D \times [0, T_{\max}] \times \mathbb{R}^n \to \mathbb{R}$, we define the mixed sup–norm $\|p\|_{L^\infty(\mathbb{R}^n;\mathcal{G})} \triangleq \sup_{x\in\mathbb{R}^n} \big\| \, (\theta,t) \mapsto p(\theta,t,x) \big\|_{\mathcal{G}}$.

A.2 Proofs organization

The proofs are structured as follows.

1. **Validity of confidence intervals (Section A.3)**: First, we establish the validity of the confidence intervals.
2. **Safety and reset guarantees (Section A.4)**: We prove that controls chosen by the algorithm maintain safety and reset properties.
3. **Sample complexity bounds (Section A.5)**: Next, we analyze the sample complexity required to achieve desired accuracy.
4. **Learning rates for safety, reset, and density estimates at fixed (θ, t) (Section A.6):** We derive learning rates, under Sobolev regularity, for estimating the safety and reset levels, as well as the state density, evaluated at fixed (θ, t) pairs.
5. **Safe learning of controlled Sobolev dynamics (Section A.7)**: Finally, we establish complete guarantees for safely learning controlled dynamics with Sobolev regularity.

A.3 Proof of the validity of confidence intervals

Assumption (A4) (Attainability)**.** *There exists a bounded and continuous reproducing kernel k defined on $D \times [0, T_{\max}]$, with associated RKHS $\mathcal{G}$, such that the safety and reset level functions satisfy*

$$s, r \in \mathcal{G}. \tag{20}$$

This assumption ensures that the safety and reset level functions can be represented in a suitable function space for estimation. In particular, it encodes prior knowledge; for example, if the functions are known to lie in a Sobolev (Hilbert) space H^m, one may choose a Sobolev kernel of order m. This is a standard assumption in the literature on kernel methods. In many practical applications, such as industrial process control and robotics [39; 51], SDEs with smooth coefficients produce smooth probability densities [17], ensuring that $s(\theta, t)$ is smooth and can be accurately represented by Gaussian or Sobolev kernels. Therefore, this assumption is mild in practice.

This lemma establishes the relevance of the defined confidence intervals.

Lemma A.1 (Validity of confidence intervals)**.** *Under Assumptions (A1), (A2), and (A4), for any $(\theta, t) \in D \times [0, T_{\max}]$ and $\lambda > 0$,*

$$|\hat{s}_N(\theta,t) - s(\theta,t)| \le \beta_N^s \sigma_N(\theta,t), \tag{21}$$
$$|\hat{r}_N(\theta,t) - r(\theta,t)| \le \beta_N^r \sigma_N(\theta,t), \tag{22}$$

where $\beta_N^s \triangleq \lambda^{-1} N^{-1/2} \max_{i\in[\![1,N]\!]} |\hat{s}_{\theta_i,t_i} - s(\theta_i,t_i)| + \|s\|_{\mathcal{G}}$, $\beta_N^r \triangleq \lambda^{-1} N^{-1/2} \max_{i\in[\![1,N]\!]} |\hat{r}_{\theta_i,t_i} - r(\theta_i,t_i)| + \|r\|_{\mathcal{G}}$.

Proof. Without loss of generality, we provide the detailed proof only for s_N, since the proof for r_N is entirely analogous.

We start by recalling the definitions

$$\hat{s}_N(\theta,t) \triangleq \hat{S}^*(K+N\lambda I)^{-1}k(\theta,t), \tag{23}$$

$$\sigma_N^2(\theta,t) \triangleq k((\theta,t),(\theta,t)) - k(\theta,t)^*(K+N\lambda I)^{-1}k(\theta,t), \tag{24}$$

where $k(\theta) \triangleq (k((\theta,t),(\theta_i,t_i)))_{i=1}^N$, $K \triangleq (k((\theta_i,t_i),(\theta_j,t_j)))_{i,j=1}^N$, and $\hat{S} = (\hat{s}_{\theta_i,t_i})_{i=1}^N$.

We define the feature map $\phi : (\theta,t) \in D \times [0,T_{\max}] \mapsto k((\theta,t),\cdot) \in \mathcal{G}$, and the operators

$$\Phi \triangleq [\phi(\theta_1,t_1),\dots,\phi(\theta_N,t_N)] \in \mathcal{G} \otimes \mathbb{R}^N, \tag{25}$$

$$\hat{C} \triangleq \frac{1}{N}\sum_{i=1}^N \phi(\theta_i,t_i)\otimes\phi(\theta_i,t_i), \tag{26}$$

such that $\hat{C} = \frac{1}{N}\Phi\Phi^*$, $K = \Phi^*\Phi$, $k((\theta,t),(\theta',t')) = \langle \phi(\theta,t),\, \phi(\theta',t')\rangle_{\mathcal{G}}$, and $k(\theta,t) = \Phi^*\phi(\theta,t)$.

Then,

$$\hat{s}_N(\theta) \triangleq \hat{S}^*(K+N\lambda I_{\mathbb{R}^N\otimes\mathbb{R}^N})^{-1}k(\theta,t), \tag{27}$$

$$= \hat{S}^*(\Phi^*\Phi+N\lambda I_{\mathbb{R}^N\otimes\mathbb{R}^N})^{-1}\Phi^*\phi(\theta,t), \tag{28}$$

$$= \hat{S}^*\Phi^*(\Phi\Phi^*+N\lambda I_{\mathcal{G}\otimes\mathcal{G}})^{-1}\phi(\theta,t), \tag{29}$$

$$= N^{-1}\hat{S}^*\Phi^*(\hat{C}+\lambda I_{\mathcal{G}\otimes\mathcal{G}})^{-1}\phi(\theta,t), \tag{30}$$

using the push-through equality $(I+AB)^{-1}A = A(I+BA)^{-1}$ for any conformal operators A, B.

Moreover,

$$\sigma_N^2(\theta) \triangleq k((\theta,t),(\theta,t)) - k(\theta,t)^*(K+N\lambda I_{\mathbb{R}^N\otimes\mathbb{R}^N})^{-1}k(\theta,t) \tag{31}$$

$$= \phi(\theta,t)^*(I_{\mathcal{G}\otimes\mathcal{G}} - \Phi(\Phi^*\Phi+N\lambda I_{\mathbb{R}^N\otimes\mathbb{R}^N})^{-1}\Phi^*)\phi(\theta,t) \tag{32}$$

$$= \phi(\theta,t)^*(I_{\mathcal{G}\otimes\mathcal{G}} - N\hat{C}(N\hat{C}+N\lambda I_{\mathcal{G}\otimes\mathcal{G}})^{-1})\phi(\theta,t) \tag{33}$$

$$= \lambda\phi(\theta,t)^*(\hat{C}+\lambda I_{\mathcal{G}\otimes\mathcal{G}})^{-1}\phi(\theta,t) \tag{34}$$

$$= \lambda\|\hat{C}_\lambda^{-1/2}\phi(\theta,t)\|_{\mathcal{G}}^2, \tag{35}$$

again using the push-through equality.

To bound the error $|\hat{s}_N(\theta,t) - s(\theta,t)|$, we decompose it as

$$|\hat{s}_N(\theta,t) - s(\theta,t)| \le \underbrace{|\hat{s}_N(\theta,t) - s_N(\theta,t)|}_{\triangleq(A)} + \underbrace{|s_N(\theta,t) - s(\theta,t)|}_{\triangleq(B)}, \tag{36}$$

where $S \triangleq (s(\theta_i,t_i))_{i=1}^N$, and $s_N(\theta,t) \triangleq N^{-1}S^*\Phi^*(\hat{C}+\lambda)^{-1}\phi(\theta,t)$.

For the first term (A), we have

$$(A) = N^{-1}|(\hat{S}-S)^*\Phi^*\hat{C}_\lambda^{-1}\phi(\theta,t)| \tag{37}$$

$$\le N^{-1}\|\hat{S}-S\|_{\mathbb{R}^N}\|\Phi^*\hat{C}_\lambda^{-1}\phi(\theta,t)\|_{\mathcal{G}} \tag{38}$$

$$\le N^{-1/2}\max_{i\in[\![1,N]\!]}|\hat{s}_{\theta_i,t_i} - s(\theta_i,t_i)|\,\|\hat{C}^{1/2}\hat{C}_\lambda^{-1}\phi(\theta,t)\|_{\mathcal{G}} \tag{39}$$

$$\le N^{-1/2}\lambda^{-1}\max_{i\in[\![1,N]\!]}|\hat{s}_{\theta_i,t_i} - s(\theta_i,t_i)|\,\sigma_N(\theta,t). \tag{40}$$

From Assumption (A4), $s(\theta,t) = \langle s, \phi(\theta,t)\rangle_{\mathcal{G}}$, such that $S = \Phi^*s$, and then $s_N(\theta,t) = s^*\Phi\Phi^*(\hat{C}+\lambda)^{-1}\phi(\theta,t) = s^*\hat{C}(\hat{C}+\lambda)^{-1}\phi(\theta,t)$.

Therefore, similarly, for the second term (B), we have

$$(B) = |s^*(\hat{C}\hat{C}_\lambda^{-1} - I)\phi(\theta,t)| \tag{41}$$

$$= \lambda|s^*\hat{C}_\lambda^{-1}\phi(\theta,t)| \tag{42}$$

$$\le \lambda^{1/2}\|s\|_{\mathcal{G}}\|\hat{C}_\lambda^{-1/2}\phi(\theta,t)\|_{\mathcal{G}} \tag{43}$$

$$= \|s\|_{\mathcal{G}}\sigma_N(\theta,t). \tag{44}$$

Combining the bounds for (A) and (B), we obtain the bound for s. Similar proof yields the bound for r.

□

In the following lemma, we derive confidence intervals for the proposed estimate of the system's dynamics $p : \theta, t, x \mapsto p_\theta(t, x)$.

Lemma A.2. *Under Assumptions (A1), (A2), and (A3), for any $(\theta, t) \in D \times [0, T_{\max}]$, we have*

$$\|\hat{p}_\theta(t, \cdot) - p_\theta(t, \cdot)\|_{L^\infty(\mathbb{R}^n)} \leq \beta_N^p \sigma_N(\theta, t), \tag{45}$$

by defining $\beta_N^p \triangleq \lambda^{-1} N^{-1/2} \max_{i \in [\![1,N]\!]} \|\hat{p}_{\theta_i,t_i}(\cdot) - p_{\theta_i}(t_i, \cdot)\|_{L^\infty(\mathbb{R}^n)} + \|p\|_{L^\infty(\mathbb{R}^n;\mathcal{G})}$

Proof. Given any $(\theta, t, x) \in D \times [0, T_{\max}] \times \mathbb{R}^n$, we have

$$\hat{p}_\theta(t, x) \triangleq \sum_{i=1}^{N} \alpha_i(\theta, t) \hat{p}_{\theta_i,t_i}(x), \tag{46}$$

with $\alpha(\theta, t) = (K + N\lambda I)^{-1} k(t, \theta)$, $K = (k((\theta_i, t_i), (\theta_j, t_j)))_{i,j=1}^N$, $k(\theta, t) = (k(\theta_i, t_i))_{i=1}^N$.

We define the feature map $\phi = (\theta, t) \in D \times [0, T_{\max}] \mapsto k((\theta, t), \cdot) \in \mathcal{G}$, and the operators

$$\Phi \triangleq [\phi(\theta_1, t_1), \ldots, \phi(\theta_N, t_N)] \in \mathcal{G} \otimes \mathbb{R}^N, \tag{47}$$

$$\hat{C} \triangleq \frac{1}{N} \sum_{i=1}^{N} \phi(\theta_i, t_i) \otimes \phi(\theta_i, t_i), \tag{48}$$

such that $\hat{C} = \frac{1}{N} \Phi \Phi^*$, $K = \Phi^* \Phi$, $k((\theta, t), (\theta', t')) = \langle \phi(\theta, t),\, \phi(\theta', t') \rangle_{\mathcal{G}}$, and $k(\theta, t) = \Phi^* \phi(\theta, t)$.

With same derivations than in the proof of Lemma A.1, for any $(\theta, t, x) \in D \times [0, T_{\max}] \times \mathbb{R}^n$, we have

$$|\hat{p}_\theta(t, x) - p_\theta(t, x)| \leq (A) + (B) \tag{49}$$

with

$$(A) \leq \lambda^{-1} N^{-1/2} \max_{i \in [\![1,N]\!]} |\hat{p}_{\theta_i,t_i}(x) - p_{\theta_i}(t_i, x)|\, \sigma_N(\theta, t). \tag{50}$$

$$(B) \leq \|p_\cdot(\cdot, x)\|_{\mathcal{G}}\, \sigma_N(\theta, t). \tag{51}$$

Therefore, for any $(\theta, t, x) \in D \times [0, T_{\max}] \times \mathbb{R}^n$, we have

$$\|\hat{p}_\theta(t, \cdot) - p_\theta(t, \cdot)\|_{L^\infty(\mathbb{R}^n)} \leq \beta_N^p \sigma_N(\theta, t), \tag{52}$$

by defining $\beta_N^p \triangleq \lambda^{-1} N^{-1/2} \max_{i \in [\![1,N]\!]} \|\hat{p}_{\theta_i,t_i}(\cdot) - p_{\theta_i}(t_i, \cdot)\|_{L^\infty(\mathbb{R}^n)} + \|p\|_{L^\infty(\mathbb{R}^n;\mathcal{G})}$

□

A.4 Proof of safety and reset guarantees

A direct consequence of Lemma A.1 is the safety and reset guarantees for the method.

Lemma A.3 (Safety guarantees)**.** *Under Assumptions (A1), (A2), and (A4), the algorithm selects only safe trajectories. Namely, for any $i \in \mathbb{N}^*$, we have $s^\infty(\theta_i, T_i) \geq 1 - \varepsilon$.*

Proof. For any $i \in [\![1, N]\!]$, we have by construction $\inf_{t \in [0,T_i]} \left(\hat{s}_N(\theta_i, t) - \beta_N \sigma_N(\theta_i, t)\right) \geq 1 - \varepsilon$.

Moreover, from Lemma A.1, we have $s(\theta, t) \geq \hat{s}(\theta, t) - \beta_N \sigma_N(\theta, t)$ for any $(\theta, t) \in D \times [0, T_{\max}]$, such that

$$s^\infty(\theta_i, T_i) \triangleq \inf_{t \in [0,T_i]} s(\theta_i, t) \tag{53}$$

$$\geq \inf_{t \in [0,T_i]} \left(\hat{s}(\theta_i, t) - \beta_N \sigma_N(\theta_i, t)\right) \tag{54}$$

$$\geq 1 - \varepsilon. \tag{55}$$

□

Lemma A.4 (Reset guarantees)**.** *Under Assumptions (A1), (A2), and (A4), the algorithm selects only resetting trajectories. Namely, for any $i \in \mathbb{N}^*$, we have $r(\theta_i, T_i) \geq 1 - \xi$.*

Proof. Similar proof as for Lemma A.3. □

A.5 Proof of sample complexity of confidence intervals

Assumption (A5) (Sublinear information growth)**.** *Let the maximum information gain up to N observations be defined as*

$$\gamma_N \triangleq \max_{(\theta_i,t_i)_{i=1}^N \in (D\times[0,T_{\max}])^N} \frac{1}{2}\sum_{i=1}^{N} \log\left(1 + \frac{\sigma_i^2(\theta_i,t_i)}{\lambda N}\right). \tag{56}$$

We assume there exist constants $\alpha \in [0,1]$ and $c > 0$ such that, for all $N \in \mathbb{N}$,

$$\gamma_N \leq cN^{\alpha}. \tag{57}$$

This assumption always holds for $\alpha = 1$ when k is bounded, since γ_N is a sum of terms bounded by $\kappa = \sup_{(\theta,t)\in D\times[0,T_{\max}]} k((\theta,t),(\theta,t))$. As α decreases within $[0,1]$, the assumption becomes stricter. It quantifies the maximal total information that can be acquired about the unknown function after N observations. Specifically, when $\alpha < 1$, the sublinear growth of information gain with N reflects diminishing returns as more data points are observed. This growth rate measures the effective dimension of the learning problem, influenced by the regularity of the RKHS and the dimensionality of $D \times [0, T_{\max}]$; higher regularity of RKHS functions and lower dimensionality of $D \times [0, T_{\max}]$ lead to a faster decay in information gain.

Example 2 (Sublinear growth for common kernels)**.** *For many commonly used kernels, γ_N exhibits sublinear growth in N, which is crucial for obtaining sublinear regret bounds in bandit problems. For example, assuming a compact domain $D \subset \mathbb{R}^m$ and setting $\lambda = N^{-1}$:*

- *RBF kernel: $\gamma_N = \mathcal{O}(\log^{m+1} N)$. This logarithmic growth arises from the high smoothness of the RBF kernel, which causes a rapid reduction in uncertainty about the unknown function as more observations are collected.*
- *Matérn kernel with $\nu > 1$: $\gamma_N = \mathcal{O}(N^{m/(m+2\nu)} \log^{2m/(m+2\nu)} N)$. This sublinear growth rate is faster than that of the RBF kernel, reflecting the lower smoothness of the Matérn kernel.*

We refer the reader to Seeger et al. [59]; Srinivas et al. [54]; Vakili et al. [60] for additional examples and their associated proofs, which show that the growth rate of γ_N depends on the eigenvalue decay in the Mercer expansion of the kernel, when available.

This lemma establishes the sample complexity of the proposed confidence intervals.

Lemma A.5 (Sample complexity of confidence intervals)**.** *Under Assumptions (A1), (A2), and (A5), for any $\eta > 0$, considering the stopping condition $\max_{(\theta,t,T)\in\Gamma_N} \sigma_N(\theta,t) < \eta$, then the proposed method stops in $N = c\eta^{-\frac{2}{1-\alpha}}$ steps where $c > 0$ is a constant that does not depend on N and η.*

Proof. Hence, under Assumption (A5), when the algorithm stops we have

$$N\eta^2 \leq \sum_{i=1}^{N} \sigma_i^2(\theta_i,t_i) \leq \frac{1}{2}\sum_{i=1}^{N} \log(1+\sigma_i^2(\theta_i,t_i)) \leq \gamma_N \leq cN^{\alpha}, \tag{58}$$

using $x \leq 2\log(1+x)$ for $x \in [0,1]$, and overloading the constant $c > 0$. Hence, $N \leq c^{-1}\eta^{-\frac{2}{1-\alpha}}$. □

A.6 Proof of learning rates for system maps estimation at fixed (θ, t)

This lemma provides a bound on the density estimation error at the selected points (θ_i, t_i).

Lemma A.6 (Density estimation learning rates)**.** *For each $i \in [\![1, N]\!]$, define $p_i \triangleq p_{\theta_i}(t_i, \cdot)$, and*

$$\hat{p}_i \triangleq \hat{p}_{\theta_i, t_i} \triangleq \frac{1}{Q} \sum_{j=1}^{Q} \rho_R(x - X_{\theta_i}(w_j, t_i)), \tag{59}$$

where $\rho_R(x) \triangleq R^{n/2}\|x\|^{-n/2} B_{n/2}(2\pi R\|x\|)$, $R > 0$, and $B_{n/2}$ is the Bessel J function of order $n/2$.

Under Assumptions (A1)-(A2), assume that $\sup_{(\theta,t)\in D\times[0,T_{\max}]} \|p_\theta(t,\cdot)\|_{H^\nu} < \infty$ for $\nu > n/2$. Set $R = Q^{\frac{1}{n+2\nu}}$. Then, with probability at least $1-\delta$, the following holds

$$\max_{i\in[\![1,N]\!]} \|\hat{p}_i - p_i\|_{L^\infty(\mathbb{R}^n)} \le c \log(4N/\delta)^{1/2} Q^{\frac{n-2\nu}{2n+4\nu}},^{2} \tag{60}$$

for some constant $c > 0$ independent of N, Q, δ.

Proof. Fix any $\varepsilon > 0$. The Sobolev embedding $H^{n/2+\varepsilon}(\mathbb{R}^n) \hookrightarrow L^\infty(\mathbb{R}^n)$ gives

$$\|\hat{p}_i - p_i\|_{L^\infty(\mathbb{R}^n)} \;\le\; c_\varepsilon \, \|\hat{p}_i - p_i\|_{H^{n/2+\varepsilon}(\mathbb{R}^n)}, \qquad i = 1, \dots, N,$$

where $c_\varepsilon > 0$ depends only on n, ε.

Following steps 3 and 4 of Theorem 5.3 in Bonalli and Rudi [17], for $\nu > n/2 + \varepsilon$ we establish

$$\begin{aligned} \max_{i\in[\![1,N]\!]} \|\hat{p}_i - p_i\|_{H^{n/2+\varepsilon}} \le \max_{i\in[\![1,N]\!]} \|p_i\|_{H^\nu} R^{\frac{n}{2}+\varepsilon-\nu} \\ + 2^{n/2+\varepsilon} R^{\frac{n}{2}+\varepsilon} 3 (V_n R)^{\frac{n}{2}} \log(4N/\delta)^{1/2} Q^{-1/2}. \end{aligned} \tag{61}$$

where V_n is the volume of the n-dimensional unit ball.

Under the assumption that $\sup_{(\theta,t)\in D\times[0,T_{\max}]} \|p_\theta(t,\cdot)\|_{H^\nu} < \infty$, we have in particular $\max_{i\in[\![1,N]\!]} \|p_i\|_{H^\nu} < \infty$. Hence, with probability at least $1-\delta$,

$$\max_{i\in[\![1,N]\!]} \|\hat{p}_i - p_i\|_{H^{n/2+\varepsilon}} \;\le\; c_\varepsilon \Big(R^{\frac{n}{2}+\varepsilon-\nu} \;+\; R^{n+\varepsilon} \left[\log(4N/\delta)\right]^{1/2} Q^{-1/2} \Big), \tag{62}$$

for some $c_\varepsilon > 0$ independent of N, Q, R, δ (but possibly depending on n, ν, ε).

Finally, setting $R = Q^{\frac{1}{n+2\nu}}$ balances the two terms in (62) and yields

$$\max_{i\in[\![1,N]\!]} \|\hat{p}_i - p_i\|_{H^{n/2+\varepsilon}} \le c [\log(4N/\delta)]^{1/2} \, Q^{\frac{n-2\nu+2\varepsilon}{2n+4\nu}}, \tag{63}$$

for some constant $c > 0$ that does not depend on N, Q, δ.

□

This lemma bounds the pointwise estimation errors of the safety and reset levels at the selected points (θ_i, t_i) in terms of the L^∞-norm error of the corresponding density estimations $\hat{p}_{\theta_i}(t_i, \cdot)$.

Lemma A.7. *Under Assumptions (A1)-(A2), we have*

$$\max_{i\in[\![1,N]\!]} |\hat{s}_{\theta_i,t_i} - s(\theta_i, t_i)| \le V_s \max_{i\in[\![1,N]\!]} \|\hat{p}_i - p_i\|_{L^\infty}, \tag{64}$$

$$\max_{i\in[\![1,N]\!]} |\hat{r}_{\theta_i,t_i} - r(\theta_i, t_i)| \le V_r \max_{i\in[\![1,N]\!]} \|\hat{p}_i - p_i\|_{L^\infty}, \tag{65}$$

where $p_i \triangleq p_{\theta_i}(t_i, \cdot)$, $V_s \triangleq \int_{\mathbb{R}^n} \mathbb{1}_{\{g(x)\ge 0\}} dx$, and $V_r \triangleq \int_{\mathbb{R}^n} \mathbb{1}_{\{h(x)\ge 0\}} dx$.

Proof. For any $i \in [\![1, N]\!]$, using Hölder's inequality, we have

$$|\hat{s}_{\theta_i,t_i} - s(\theta_i, t_i)| \le V_s \|\hat{p}_i - p_i\|_{L^\infty}, \tag{66}$$

where $p_i \triangleq p_{\theta_i}(t_i, \cdot)$, and $V_s \triangleq \int_{\mathbb{R}^n} \mathbb{1}_{\{g(x)\ge 0\}} dx$.

Similarly, $|\hat{r}_{\theta_i,t_i} - r(\theta_i, t_i)| \le V_r \|\hat{p}_i - p_i\|_{L^\infty}$ where $V_r \triangleq \int_{\mathbb{R}^n} \mathbb{1}_{\{h(x)\ge 0\}} dx$.

□

[2] Throughout, exponents involving n should be interpreted as $n + \varepsilon$ for arbitrarily small $\varepsilon > 0$.

A.7 Proof of safe learning of controlled Sobolev dynamics

We conclude the following theorem from all previously established lemmas.

Theorem A.8 (Safely learning controlled Sobolev dynamics)**.** *Let $\eta > 0$, and assume Assumptions (A1)–(A3) hold. Set $R = Q^{1/(n+2\nu)}$. Then there exist constants $c_1, \dots, c_5 > 0$, independent of N, Q, δ, η, such that if*

$$c_1 \log(4N/\delta)^{1/2} Q^{\frac{n-2\nu}{2n+4\nu}} \leq N^{-1/2},$$

then the stopping condition $\max_{(\theta,t,T)\in\Gamma_N} \sigma_N(\theta,t) < \eta$ is satisfied after at most $N \leq c_2 \eta^{-2/(1-\alpha)}$ iterations for any $\alpha > (m+1)/(m+1+2\nu)$. Moreover:

- ***(Safety):** All selected triples (θ_i, t_i, T_i) satisfy $s^\infty(\theta_i, T_i) \geq 1-\varepsilon$ and $r(\theta_i, T_i) \geq 1-\xi$. The final set Γ_N includes only controls meeting these thresholds and can thus serve as a certified safe set for deployment.*
- ***(Estimation accuracy):** For all $(\theta, t, T) \in \Gamma_N$,*

$$\|\hat{p}_\theta(t,\cdot) - p_\theta(t,\cdot)\|_\infty \leq c_3\eta, \quad |\hat{s}_N(\theta,t) - s(\theta,t)| \leq c_4\eta, \quad |\hat{r}_N(\theta,t) - r(\theta,t)| \leq c_5\eta.$$

Proof. From Lemma A.1 and Lemma A.2,for any $(\theta,t) \in D \times [0, T_{\max}]$ and $\lambda > 0$,

$$|\hat{s}_N(\theta,t) - s(\theta,t)| \leq \beta_N^s \sigma_N(\theta,t), \tag{67}$$
$$|\hat{r}_N(\theta,t) - r(\theta,t)| \leq \beta_N^r \sigma_N(\theta,t), \tag{68}$$
$$\|\hat{p}_\theta(t,\cdot) - p_\theta(t,\cdot)\|_{L^\infty(\mathbb{R}^n)} \leq \beta_N^p \sigma_N(\theta,t), \tag{69}$$

where $\beta_N^s \triangleq \lambda^{-1} N^{-1/2} \max_{i\in[\![1,N]\!]} |\hat{s}_{\theta_i,t_i} - s(\theta_i,t_i)| + \|s\|_{\mathcal{G}}$, $\beta_N^r \triangleq \lambda^{-1} N^{-1/2} \max_{i\in[\![1,N]\!]} |\hat{r}_{\theta_i,t_i} - r(\theta_i,t_i)| + \|r\|_{\mathcal{G}}$, and $\beta_N^p \triangleq \lambda^{-1} N^{-1/2} \max_{i\in[\![1,N]\!]} \|\hat{p}_{\theta_i,t_i}(\cdot) - p_{\theta_i}(t_i,\cdot)\|_{L^\infty(\mathbb{R}^n)} + \|p\|_{L^\infty(\mathbb{R}^n;\mathcal{G})}$.

Then, from Lemma A.7, we have

$$\max_{i\in[\![1,N]\!]} |\hat{s}_{\theta_i,t_i} - s(\theta_i,t_i)| \leq V_s \max_{i\in[\![1,N]\!]} \|\hat{p}_i - p_i\|_{L^\infty}, \tag{70}$$
$$\max_{i\in[\![1,N]\!]} |\hat{r}_{\theta_i,t_i} - r(\theta_i,t_i)| \leq V_r \max_{i\in[\![1,N]\!]} \|\hat{p}_i - p_i\|_{L^\infty}, \tag{71}$$

However, from Lemma A.6, with probability at least $1-\delta$, we have

$$\max_{i\in[\![1,N]\!]} \|\hat{p}_i - p_i\|_{L^\infty(\mathbb{R}^n)} \leq c \log(4N/\delta)^{1/2} Q^{\frac{n-2\nu}{2n+4\nu}}, \tag{72}$$

such that there exists $c_1 > 0$ independent of N, Q, δ, η such that if

$$c_1 \log(4N/\delta)^{1/2} Q^{\frac{n-2\nu}{2n+4\nu}} \leq (V_s \vee V_r)^{-1} N^{-1/2}, \tag{73}$$

then

$$\max_{i\in[\![1,N]\!]} \|\hat{p}_i - p_i\|_{L^\infty(\mathbb{R}^n)} \leq N^{-1/2}. \tag{74}$$

Therefore, from Lemma A.7, we have, if $\lambda = N^{-1}$, for any $i \in [\![1,N]\!]$,

$$\beta_N^s \triangleq N^{1/2} \max_{j\in[\![1,N]\!]} |\hat{s}_{\theta_j,t_j} - s_{\theta_j,t_j}| + \|s\|_{\mathcal{G}} \tag{75}$$
$$\leq 1 + \|s\|_{\mathcal{G}}. \tag{76}$$

Similarly, we have $\beta_N^r \leq 1 + \|r\|_{\mathcal{G}}$ and $\beta_N^p \leq 1 + \|p\|_{L^\infty(\mathbb{R}^n;\mathcal{G})}$.

Assumption (A3) simultaneously guarantees $\|p\|_{L^\infty(\mathbb{R}^n;\mathcal{G})} \triangleq \sup_{x\in\mathbb{R}^n} \big\| (\theta,t) \mapsto p(\theta,t,x) \big\|_{\mathcal{G}} < +\infty$, $\sup_{(\theta,t)\in D\times[0,T_{\max}]} \|p_\theta(t,\cdot)\|_{H^\nu} < \infty$ with $\nu > n/2$. Furthermore, considering a kernel that induces a Sobolev RKHS $\mathcal{G}$ of order ν, Assumption (A4) holds true, and Assumption (A5) holds for any $\alpha > (m+1)/(m+1+2\nu)$ as mentioned in the examples of Assumption (A5).

Therefore, from Lemma A.3, and Lemma A.4, we have that:

- All selected controls are safe, i.e., $s^\infty(\theta_i, T_i) \geq 1 - \varepsilon$ for any $i \in \mathbb{N}^*$.
- All selected controls ensure reset, i.e., $r(\theta_i, T_i) \geq 1 - \xi$ for any $i \in \mathbb{N}^*$.

Moreover, Lemma A.5 ensures that for any $\alpha > (m+1)/(m+1+2\nu)$, with probability at least $1-\delta$, the stopping condition

$$\max_{\theta,t,T \in \Gamma_N} \sigma_N(\theta, t) < \eta, \tag{77}$$

is reached for $N \leq c_2 \eta^{-\frac{2}{1-\alpha}}$ where c_2 does not depend on N, Q, δ, η.

□

B Implementation details

This section details the implementation of the method described in Section 4, available on GitHub (lmotte/dynamics-safe-learn) as an open-source Python library. We detail all computational steps, including vectorized implementations using Python libraries such as NumPy for efficiency. Additionally, we outline the computational complexity of each step to provide insights into scalability.

B.1 System estimation

Density estimation. For each data point (θ_i, t_i, T_i), the density is estimated with

$$\hat{p}_{\theta_i,t_i}(x) = \frac{1}{Q} \sum_{j=1}^{Q} \rho_R(x - X_{u_{\theta_i}}(w_j^i, t_i)),$$

where Q is the number of samples generated for each control θ_i and time t_i, $\rho_R(x)$ is a kernel density function, typically defined as $\rho_R(x) = R^{n/2}\|x\|^{-n/2} J_{n/2}(2\pi R\|x\|)$, with $R > 0$ and $J_{n/2}$ the Bessel function of order $n/2$, $X_{u_{\theta_i}}(w_j^i, t_i)$ are the system trajectories generated under control u_{θ_i}.

Computational complexity. Evaluating $\hat{p}_{\theta_i,t_i}(x)$ requires $\mathcal{O}(Q)$ operations per data point. This step is trivially parallelizable across data points, since all kernel evaluations are independent.

Algorithm 1: DensityEstimation($\{x_j\}_{j=1}^Q, \rho_R$)

Input: Q trajectory samples $\{x_j\}_{j=1}^Q$ at (θ_i, t_i), kernel ρ_R
Output: density estimator $\hat{p}_{\theta_i,t_i} : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$

1 **if** x *is queried* **then return** $\hat{p}_{\theta_i,t_i}(x) \leftarrow \frac{1}{Q} \sum_{j=1}^{Q} \rho_R(x - x_j)$

Probability Computation ($\hat{s}, \hat{r}$). The vectors $\hat{P}$, $\hat{S}$, and $\hat{R}$ are constructed from the observed data points $\{(\theta_i, t_i, T_i)\}_{i=1}^N$ as follows:

$$\hat{P}(\cdot) = \begin{bmatrix} \hat{p}_{\theta_1}(t_1, \cdot) \\ \hat{p}_{\theta_2}(t_2, \cdot) \\ \vdots \\ \hat{p}_{\theta_N}(t_N, \cdot) \end{bmatrix}, \quad \hat{S} = \begin{bmatrix} \hat{s}_{\theta_1,t_1} \\ \hat{s}_{\theta_2,t_2} \\ \vdots \\ \hat{s}_{\theta_N,t_N} \end{bmatrix}, \quad \hat{R} = \begin{bmatrix} \hat{r}_{\theta_1,t_1} \\ \hat{r}_{\theta_2,t_2} \\ \vdots \\ \hat{r}_{\theta_N,t_N} \end{bmatrix}.$$

The vectors $\hat{S}$ and $\hat{R}$ are computed by integrating the densities over the safe and reset regions, respectively:

$$\hat{s}_{\theta_i,t_i} = \int_{\{x \in \mathbb{R}^n : g(x) \geq 0\}} \hat{p}_{\theta_i,t_i}(x)\, dx,$$

$$\hat{r}_{\theta_i,t_i} = \int_{\{x\in\mathbb{R}^n : h(x)\geq 0\}} \hat{p}_{\theta_i,t_i}(x)\,dx.$$

These integrals are approximated using Monte Carlo integration. If we can sample $x_k \sim \hat{p}_{\theta_i,t_i}$, then

$$\hat{s}_{\theta_i,t_i} \approx \frac{1}{Q'}\sum_{k=1}^{Q'} \mathbb{1}\{g(x_k)\geq 0\}, \qquad \hat{r}_{\theta_i,t_i} \approx \frac{1}{Q'}\sum_{k=1}^{Q'} \mathbb{1}\{h(x_k)\geq 0\}.$$

Alternatively, using trajectory samples $\{X_{u_{\theta_i}}(w_j^i, t_i)\}_{j=1}^{Q}$,

$$\hat{s}_{\theta_i,t_i} \approx \frac{1}{Q}\sum_{j=1}^{Q} \mathbb{1}\{g(X_{u_{\theta_i}}(w_j^i,t_i))\geq 0\}, \qquad \hat{r}_{\theta_i,t_i} \approx \frac{1}{Q}\sum_{j=1}^{Q} \mathbb{1}\{h(X_{u_{\theta_i}}(w_j^i,t_i))\geq 0\}.$$

Here $\mathbb{1}\{\cdot\}$ denotes the indicator function.

Computational complexity. Constructing $\hat{P}$ involves N density evaluations, each costing $\mathcal{O}(Q)$, for a total of $\mathcal{O}(NQ)$. For $\hat{S}$ and $\hat{R}$, Monte Carlo based on samples from $\hat{p}$ costs $\mathcal{O}(NQ')$, while using trajectory samples costs $\mathcal{O}(NQ)$. Overall, constructing $(\hat{P}, \hat{S}, \hat{R})$ costs $\mathcal{O}(NQ)$ (trajectory MC) or $\mathcal{O}(N(Q+Q'))$ (using both). All computations are trivially parallelizable across evaluation points.

Algorithm 2: ComputeProbabilities($\hat{p}_{\theta_i,t_i}$, g, h, Q', `mode`)

Input: density estimator $\hat{p}_{\theta_i,t_i}$, constraint functions g, h, number of MC samples Q', mode $\in \{\texttt{density}, \texttt{trajectory}\}$
Output: probabilities $(\hat{s}_{\theta_i,t_i},\ \hat{r}_{\theta_i,t_i})$

if *mode* $=$ *density* **then**
  Sample $\{x_k\}_{k=1}^{Q'}$ **from** $\hat{p}_{\theta_i,t_i}$
  $\hat{s} \leftarrow \frac{1}{Q'}\sum_{k=1}^{Q'} \mathbf{1}\{g(x_k)\geq 0\}$
  $\hat{r} \leftarrow \frac{1}{Q'}\sum_{k=1}^{Q'} \mathbf{1}\{h(x_k)\geq 0\}$
else `// trajectory`
  Reuse trajectory samples $\{x_j\}_{j=1}^{Q}$ at (θ_i, t_i)
  $\hat{s} \leftarrow \frac{1}{Q}\sum_{j=1}^{Q} \mathbf{1}\{g(x_j)\geq 0\}$
  $\hat{r} \leftarrow \frac{1}{Q}\sum_{j=1}^{Q} \mathbf{1}\{h(x_j)\geq 0\}$
return $(\hat{s}, \hat{r})$

Kernel-based estimation of system maps. Given N observed data points $\{(\theta_i, t_i, T_i)\}_{i=1}^{N}$, the Gram matrix K is constructed as

$$K_{ij} = k((\theta_i,t_i),(\theta_j,t_j)),$$

where k is the kernel function. For a new input (θ, t), the estimates for the system dynamics, safety, and reset functions are

$$\hat{p}_\theta(t,x) = \hat{P}(x)(K+N\lambda I)^{-1}k(\theta,t),$$
$$\hat{s}_N(\theta,t) = \hat{S}(K+N\lambda I)^{-1}k(\theta,t),$$
$$\hat{r}_N(\theta,t) = \hat{R}(K+N\lambda I)^{-1}k(\theta,t),$$

where $k(\theta,t) = [k((\theta,t),(\theta_i,t_i))]_{i=1}^N$, λ is a regularization parameter, $\hat{P}(\cdot) \triangleq (\hat{p}_{\theta_i,t_i}(\cdot))_{i=1}^N$, $\hat{S} = (\hat{s}_{\theta_i,t_i})_{i=1}^N$, $\hat{R} = (\hat{r}_{\theta_i,t_i})_{i=1}^N$.

The predictive uncertainty for (θ,t) is computed as

$$\sigma_N^2(\theta,t) = k((\theta,t),(\theta,t)) - k(\theta,t)^*(K + N\lambda I)^{-1}k(\theta,t).$$

Computational complexity. Gram matrix construction requires $\mathcal{O}(N^2)$ operations, matrix inversion involves $\mathcal{O}(N^3)$ operations. Evaluating $\hat{p}_N(\theta,t)$, $\hat{s}_N(\theta,t)$, $\hat{r}_N(\theta,t)$, or $\sigma_N^2(\theta,t)$, costs $\mathcal{O}(N^2)$ per prediction.

Algorithm 3: FitKernelMaps($\{(\theta_i,t_i)\}_{i=1}^N$, $\hat{P}(\cdot)$, $\hat{S}$, $\hat{R}$, k, λ)

Input: inputs $\{(\theta_i,t_i)\}_{i=1}^N$, targets $\hat{P}(\cdot), \hat{S}, \hat{R}$, kernel k, regularization λ
Output: query operators for $\hat{p}_\theta(t,\cdot)$, $\hat{s}_N(\theta,t)$, $\hat{r}_N(\theta,t)$, $\sigma_N^2(\theta,t)$

Build K with $K_{ij} \leftarrow k((\theta_i,t_i),(\theta_j,t_j))$; set $K_\lambda \leftarrow K + N\lambda I$

Compute and store K_λ^{-1}

Precompute: $\alpha_S \leftarrow K_\lambda^{-1}\hat{S}, \quad \alpha_R \leftarrow K_\lambda^{-1}\hat{R}$

Query at (θ,t) (and optionally at x):

1. $k \leftarrow [\,k((\theta,t),(\theta_i,t_i))\,]_{i=1}^N$
2. $z \leftarrow K_\lambda^{-1}k$ `// used for variance and density map`
3. $\hat{s}_N(\theta,t) \leftarrow k^\top \alpha_S$
4. $\hat{r}_N(\theta,t) \leftarrow k^\top \alpha_R$
5. $\sigma_N^2(\theta,t) \leftarrow k((\theta,t),(\theta,t)) - k^\top z$
6. **If density at x is requested:** build $\hat{P}(x) \leftarrow [\hat{p}_{\theta_1,t_1}(x),\ldots,\hat{p}_{\theta_N,t_N}(x)]^\top$ and return $\hat{p}_\theta(t,x) \leftarrow \hat{P}(x)^\top z$

B.2 Safe sampling

Feasibility criteria. The feasibility criteria for safe sampling are defined as:

$$\mathrm{LCB}_\mathrm{N}^\mathrm{s}(\theta,T) \triangleq \inf_{t\in[0,T]} \left(\hat{s}_N(\theta,t) - \beta_N^s \sigma_N(\theta,t)\right),$$

$$\mathrm{LCB}_\mathrm{N}^\mathrm{r}(\theta,T) \triangleq \hat{r}_N(\theta,T) - \beta_N^r \sigma_N(\theta,T),$$

where $\beta_N^s, \beta_N^r > 0$ are confidence parameters. The safe-resettable feasible set is:

$$\Gamma_N = \left\{(\theta,t,T) \in D \times [0,T_{\max}]^2 \;\middle|\; t \le T,\ \ \mathrm{LCB}_\mathrm{N}^\mathrm{s}(\theta,T) \ge 1-\varepsilon,\ \ \mathrm{LCB}_\mathrm{N}^\mathrm{r}(\theta,T) \ge 1-\xi\right\} \cup \Gamma_0,$$

with

$$\Gamma_0 = \left\{(\theta,t,T_{\max}) \in D \times [0,T_{\max}]^2 \;\middle|\; (\theta,t) \in S_0 \cap R_0\right\}.$$

Sampling rule. The next control, time, and trajectory are selected by solving:

$$(\theta_{N+1}, t_{N+1}, T_{N+1}) = \operatorname*{arg\,max}_{(\theta,t,T)\in\Gamma_N} \sigma_N(\theta,t),$$

where

$$\sigma_N^2(\theta,t) = k((\theta,t),(\theta,t)) - k(\theta,t)^*(K + N\lambda I)^{-1}k(\theta,t).$$

Stopping rule. Exploration terminates when:

$$\max_{(\theta,t,T)\in\Gamma_N} \sigma_N(\theta,t) < \eta,$$

where $\eta > 0$ is the user-defined uncertainty threshold.

Computational complexity.

- **Set construction (Γ_N):** Evaluating $\mathrm{LCB}_{\mathrm{N}}^{\mathrm{s}}$ and $\mathrm{LCB}_{\mathrm{N}}^{\mathrm{r}}$ involves $\mathcal{O}(N^2)$ operations for each candidate point. For M candidate points, constructing Γ_N costs $\mathcal{O}(MN^2)$.
- **Uncertainty evaluation:** Evaluating σ_N involves matrix-vector multiplications and inversions, where Gram matrix construction costs $\mathcal{O}(N^2)$, matrix inversion costs $\mathcal{O}(N^3)$, and per-query uncertainty evaluation costs $\mathcal{O}(N^2)$.
- **Optimization ($\arg\max$):**
  - For discretization over M candidates: $\mathcal{O}(MN^2)$,
  - For gradient-based optimization: $\mathcal{O}(kN^2)$, where k is the number of optimization iterations, using a nonlinear constrained solver (e.g. SQP or interior-point). Each iteration involves computing the gradient of $\sigma_N(\theta, t)$, costing $\mathcal{O}(N^2)$.

Efficient sampling algorithm. In practice, the sampling process can be improved to reduce computational costs by focusing on promising regions and avoiding unnecessary evaluation of low-uncertainty points: (1) threshold-based filtering: select any candidate where $\sigma_N(\theta, t) > \eta$ to avoid costly global optimization while maintaining guarantees; (2) exclude evaluated points: skip candidates where $\sigma_N(\theta, t) < \eta$, assuming the uncertainty is non-increasing (true for $\lambda = 1/N$); (3) localized sampling: restrict Γ_N to points near the initial safe set and previously selected points. Algorithm 4 implements a region-growing strategy that encourages local exploration around previously selected safe points.

Algorithm 4: Efficient sampling algorithm

Input: Initial safe-resettable set $S_0 \cap R_0$, threshold η, feasible set Γ_N
Output: Updated sets $\mathcal{P}_k$, $\mathcal{A}_k$, and the selected candidate (if found)

Initialization:
Set $\mathcal{P}_0 = S_0 \cap R_0$ and $\mathcal{A}_0 = \emptyset$
Define the feasible set using a localized search region:

$$\Gamma^k = \Gamma_N \cap \{(\theta, t, T) \mid d((\theta, t), \mathcal{P}_k) \leq r_k\} \setminus \mathcal{A}_k,$$

where $d((\theta, t), \mathcal{P}_k)$ denotes the minimum Euclidean distance from (θ, t) to any point in $\mathcal{P}_k$.
Iterations:
for $k = 0, 1, 2, \ldots$ **do**
  foreach *candidate* $(\theta, t, T) \in \Gamma^k$ **do**
    if $\sigma_N(\theta, t) > \eta$ **then**
      Select the candidate (θ, t, T)
      Update the safe-resettable set:

$$\mathcal{P}_{k+1} = \mathcal{P}_k \cup \{(\theta, t, T)\}$$

      return $(\mathcal{P}_{k+1}, \mathcal{A}_k, (\theta, t, T))$
    end
    else
      Update the excluded set:

$$\mathcal{A}_{k+1} = \mathcal{A}_k \cup \{(\theta, t, T)\}$$

    end
  end
  Expand the radius: $r_k \to r_{k+1}$
  Recompute Γ^k
  if $\Gamma^k = \emptyset$ **then**
    return $(\mathcal{P}_k, \mathcal{A}_k, \text{None})$
  end
end

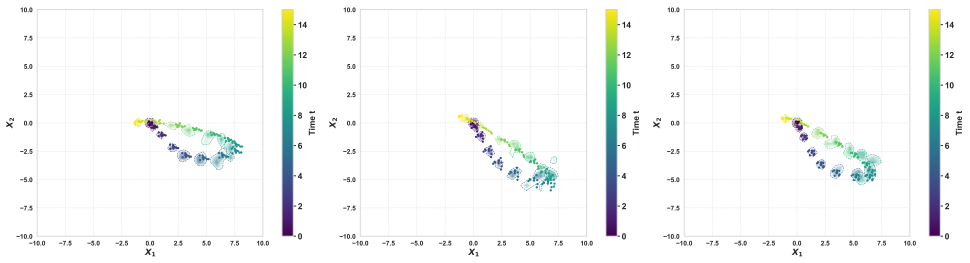


Figure 7: Predicted probability density of the trained model ($\varepsilon = \xi = 0.1$) along with 10 true trajectories, for three test controls ((-0.81 1.20), (-1.07 0.92), (-1.13 1.18)) from the known set of safe controls for this model. Predictions were computed on a spatial grid of 2500 points and times $t \in [\![0, 16]\!]$.

B.3 Role and tuning of hyperparameters

Hyperparameters critically influence the balance between exploration, safety, and computational efficiency in our method. The safety and reset thresholds (ε, ξ) directly control this balance: lower thresholds enforce stricter constraints, restricting exploration to safer regions at the expense of slower coverage, whereas higher thresholds allow broader exploration but increase risk. The confidence parameters (β_s, β_r) modulate how conservatively the safe-resettable set expands, reflecting tolerance to uncertainty in safety and reset predictions. The parameters λ and γ define the smoothness of the estimated functions, thus capturing prior knowledge about the system dynamics. Different hyperparameters may be chosen for each map: γ_{kde} and λ_{kde} for density estimation, and $\gamma_{\text{collect}}, \lambda_{\text{collect}}, \beta_{\text{collect}}$ for safety and reset predictions, as these functions may have different smoothness characteristics. Typically, hyperparameters are tuned using validation data. Parameters for density estimation ($\gamma_{\text{kde}}, \lambda_{\text{kde}}$) can be optimized after data collection by maximizing log-likelihood. In contrast, the parameters governing safety and reset exploration ($\gamma_{\text{collect}}, \lambda_{\text{collect}}, \beta_{\text{collect}}$) must be set beforehand, as they directly impact safe exploration. When prior knowledge is limited, we recommend initially conservative settings—high γ, low λ, high β, and large kernel bandwidth R—and gradually relaxing them based on data-driven insights. In our experiments, γ_{kde} and λ_{kde} were visually tuned using validation controls $(2\pi/3, -\pi/3)$, $(-2\pi/3, -\pi/3)$, and $(0, -\pi/3)$. Meanwhile, $\gamma_{\text{collect}}, \lambda_{\text{collect}}, \beta_{\text{collect}}$ were set heuristically, assuming reasonable prior smoothness estimates to minimize computational overhead. Without such prior knowledge, comprehensive hyperparameter tuning would likely demand significantly higher computational resources, as extensive parameter searches become necessary.

B.4 Computational considerations

To provide practical insight into computational requirements, we report measured execution times from experiments on a standard machine (Apple M3 Pro, 18GB RAM). Each training iteration required approximately $1.92s \pm 0.02$, totaling around 31.77 ± 0.38 minutes for 1000 iterations, based on 20 repeated runs. This includes candidate selection, trajectory simulation, computing safety and reset probabilities, and model updates at each iteration. Density predictions took approximately 10 seconds in average for computing $p(\theta, t, x)$ over a grid of 2500 x for each considered (θ, t). Although the computational times are non-trivial, they remain manageable on standard hardware for the problem sizes considered. It should be noted that several approximation methods—such as sketching for matrix inversion and online matrix inversion—as well as parallelization techniques (e.g., parallelizing the simulations) can be leveraged to alleviate the computational burden. However, exploring these techniques is beyond the scope of this paper.

C Additional experimental results

C.1 Dynamics prediction accuracy

In Figure 7, we present the predicted probability density from the model trained with $\varepsilon = \xi = 0.1$, alongside 10 true trajectories for three test controls ((-0.81 1.20), (-1.07 0.92), (-1.13 1.18)) chosen from the known set of safe controls with uncertainty below 0.1. The density is evaluated over a spatial grid of 2500 points and time steps $t \in [\![0, 16]\!]$. This visualization provides a qualitative assessment of

the model's dynamic prediction accuracy. The predicted probability distributions closely match the true dynamics, exhibiting similar means and variances over time.

C.2 Information gain over iterations

To complement the analysis of exploration behavior, we report the cumulative information gain over the course of training, for different safety and reset thresholds $\varepsilon = \xi \in \{0.1, 0.3, 0.5, +\infty\}$. Figure 8 shows how the information gain evolves as new trajectory data is collected.

We observe that larger thresholds, which allow more aggressive exploration, result in faster information acquisition. In contrast, stricter thresholds slow down exploration and yield more gradual information growth. This reflects the fundamental trade-off between exploration and safety: ensuring high-probability safety requires restricting the sampling space, particularly in regions with high model uncertainty.

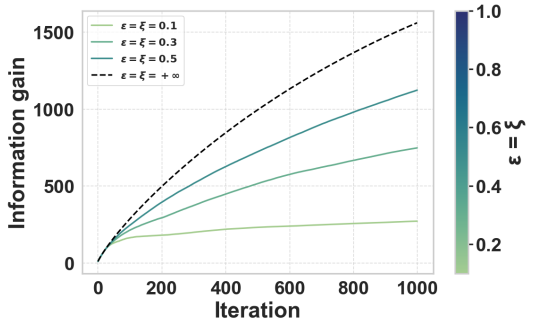


Figure 8: Cumulative information gain over iterations for various thresholds $\varepsilon = \xi \in \{0.1, 0.3, 0.5, +\infty\}$.