Safe Guaranteed Dynamics Exploration with Probabilistic Models

Manish Prajapat
ETH Zurich

Johannes Köhler
ETH Zurich

Melanie N. Zeilinger†
ETH Zurich

Andreas Krause†
ETH Zurich

Abstract

Deploying agents in the real world is inherently challenging due to their *a priori* unknown dynamics and the need for rigorous safety guarantees. Without an accurate model, the agent risks taking unsafe actions or even failing to complete their tasks. To address this problem, we introduce a notion of maximum safe dynamics learning through sufficient exploration in the space of safe policies. We propose a *pessimistically* safe framework that *optimistically* explores informative states and, despite not reaching them due to model uncertainty, ensures continuous online learning of dynamics. The framework achieves first-of-its-kind results: non-episodically learning the dynamics model sufficiently — up to an arbitrary small tolerance (subject to noise) — in a finite time, while ensuring provably safe operation throughout with high probability. Building on this, we propose an algorithm to maximize rewards while learning the dynamics *only to the extent needed* to achieve close-to-optimal performance. Unlike typical reinforcement learning (RL) methods, our approach operates online in a non-episodic setting and ensures safety throughout the learning process. We demonstrate the effectiveness of our approach in challenging domains such as autonomous car racing and drone navigation under aerodynamic effects — scenarios where safety is critical and accurate modeling is difficult.

1 Introduction

Ensuring both optimality and safety is critical for the real-world deployment of agents; however, this becomes particularly challenging when the system dynamics are unknown. In such cases, the agent must learn the dynamics online while interacting with the environment. Crucially, safety must be maintained throughout the learning process as the real-world deployments are inherently non-episodic, where resetting back to the initial state is not an option. Therefore, achieving sufficient learning for optimal behavior in a safe, online, and non-episodic manner is essential across a wide range of applications, especially in autonomous robotics [1], including car racing [2], drones, and underwater vehicles.

Related work. Model-based reinforcement learning (RL) is a well-established framework to achieve close-to-optimal performance through optimistic or sampling-based exploration [3–5]. To ensure safety, constraints under expectation are additionally enforced through Constrained Markov Decision Processes (CMDPs) [6]; however, these approaches typically ensure safety only for the final learned policy, not throughout the learning process itself [7–11]. Safety during learning is often addressed through pessimistic exploration strategies inspired by the safe Bayesian optimization (BO) literature [12–15]. These methods have been extended to optimize safe policies [16, 17]. However, they require resets during the learning process, which may not be possible in non-episodic real-world applications. Safe exploration *without* resets has been developed more recently, using techniques from model predictive control (MPC) [18]. However, this method focuses only on learning uncertain constraints, and steering the system to informative states requires that the dynamics are sufficiently well-known. The problem of safely controlling an uncertain dynamical system without resets has also been studied in the control community, especially using (learning-based) MPC [19–22]. However,

† Joint supervision. Correspondence to `manishp@ai.ethz.ch`.

2nd Workshop on Aligning Reinforcement Learning Experimentalists and Theorists (ARLET 2025).

only a few works consider actively learning dynamics [23–25], and in general, these algorithms do not guarantee that the dynamics model is learned sufficiently accurately, let alone providing optimality guarantees. In contrast, our work *uniquely addresses safe guaranteed exploration without resets, ensuring close-to-optimal performance in finite-time*. Appendix A provides a more detailed comparison to related work. To address the open problem of safe, non-episodic learning of unknown dynamics with sufficient exploration, we make the following key contributions:

Firstly, to ensure sufficient dynamics learning for any task, we introduce the notion of guaranteed exploration in the policy space, which ensures *maximum safe dynamics learning* over the constrained set. We propose a general framework that pessimistically ensures safety against model uncertainty while optimistically exploring informative states. Despite not exactly reaching the desired state, the proposed approach ensures that the agent obtains informative measurements. **Secondly**, we theoretically guarantee that the dynamics model is learned sufficiently up to a user-chosen tolerance (subject to noise) in finite time, while being provably safe throughout with high probability. **Thirdly**, building on the general framework, we propose an efficient algorithm to maximize rewards, where the dynamics model is learned only to the extent needed to achieve optimal behavior. The approach maintains safety by ensuring returnability to a known safe set, but without having to actually return, thus ensuring efficiency. We prove that the agent achieves close-to-optimal performance in finite time, while being provably safe throughout the (non-episodic) online learning process. **Finally**, we provide an efficient sampling-based implementation of our method and demonstrate its effectiveness with unknown dynamical models in the challenging safety-critical task of autonomous racing and drone navigation under aerodynamic effects. Our experiments demonstrate rapid convergence to optimal policies without compromising safety.

2 Problem setup

We consider the task of learning and controlling a nonlinear dynamical system

$$x(k+1) = \boldsymbol{f}^\star(x(k), u(k)) + \eta(k) \tag{1}$$

with state $x(k) \in \mathbb{R}^{n_x}$ and input $u(k) \in \mathbb{R}^{n_u}$. The noise $\eta(k) \in \mathcal{W} \subseteq \mathbb{R}^{n_x}$ is assumed to be bounded and conditionally σ-sub-Gaussian [26]. Here, $\boldsymbol{f}^\star : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ denotes the dynamics model which is a-priori unknown and needs to be learned using online state measurements $x(k)$. The disturbance set is given by $\mathcal{W} \coloneqq \mathcal{B}_{\bar{\eta}}$, where $\mathcal{B}_r$ is an infinity-norm ball of radius r with appropriate dimension. The system should satisfy (known) state and input constraints during closed-loop operation:

$$x(k) \in \mathcal{X}, u(k) \in \mathcal{U}, \ \forall k \in \mathbb{N}. \tag{2}$$

In our setting, the agent plans in real time (online) at time step $k \in \mathbb{N}$ for a finite horizon H, x_h denotes h predictions in the future starting with $x_0 = x(k)$ and $h \in \mathbb{N}_{[0,\mathrm{H}-1]} \coloneqq \{0, 1, \dots, \mathrm{H}-1\}$.

Objective. Our goal is to learn the unknown system (1) up to user-chosen tolerance $\epsilon > 0$, while satisfying the constraints (2) with arbitrary high-probability $1-\delta \in (0,1)$ during runtime, i.e., in a non-episodic fashion without resets (Section 3). In addition, using this guaranteed exploration framework, we aim to safely maximize rewards, i.e., to find a policy π that maximizes,

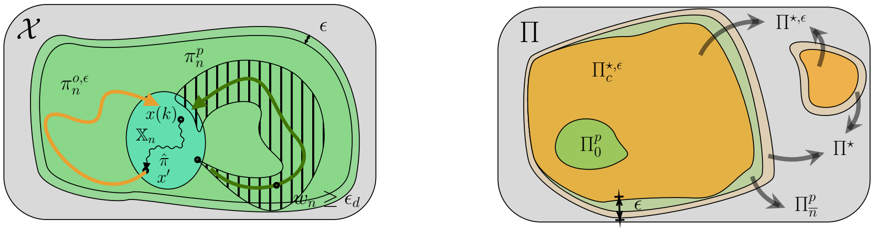
$$J(x(k), \boldsymbol{f}; \pi) \coloneqq \mathbb{E}\left[\sum_{h=0}^{\mathrm{H}-1} r(x_h, u_h) | x_0 = x(k), \pi\right], \tag{3}$$

over a finite horizon H for the (unknown) true dynamics $\boldsymbol{f} = \boldsymbol{f}^\star$ (1), where $u_h = \pi_h(x_h)$, and safety (2) must be ensured throughout the process (Section 4). We operate in MPC-style, that is, at real time k, we optimize (plan) for a prediction of finite horizon H, then execute actions and do this planning repeatedly. Without loss of generality, we assume that the reward $r : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ is non-negative, i.e., $r(\cdot,\cdot) \geq 0$. We optimize over the class of non-stationary deterministic policies $\pi \coloneqq [\pi_0, \dots \pi_{\mathrm{H}-1}]^\top \in \Pi_{\mathrm{H}}$ with $\pi_h : \mathcal{X} \to \mathcal{U}, h \in \mathbb{N}$.

2.1 Probabilistic dynamic model

We make a standard regularity assumption on the unknown dynamics $\boldsymbol{f}^\star$ [15, 17, 27, 28]:

Assumption 1 (Regularity)**.** *Each component $f_i^\star, i \in \mathbb{N}_{[1,n_x]}$ of the unknown dynamics model is an element of the Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_{k^i}$ associated with a continuous, positive definite kernel $k^i : \mathbb{R}^{n_x+n_u} \times \mathbb{R}^{n_x+n_u} \to \mathbb{R}_{\geq 0}$, with a bounded RKHS norm, i.e., $f_i^\star \in \mathcal{H}_{k^i}$ with $\|f_i^\star\|_{\mathcal{H}_{k^i}} \leq B_i < \infty$ where B_i is known. Furthermore, $k^i(x,x) \leq 1$, $\forall x \in \mathbb{R}^{n_x+n_u}$.*



(a) Optimistic and pessimistic trajectories

(b) Convergence of pessimistic policy set

Figure 1: Illustration of policy set in (a) state space and (b) policy space. In Fig. 1a, the cyan region denotes the (invariant) safe set $\mathbb{X}_n$ and the green region represents the state constraint $\mathcal{X}$. The shaded region shows the reachable set under a pessimistic policy, which starts in the safe set and returns to it while satisfying the constraints. The green curve shows an informative trajectory ensuring sampling condition (8). The orange curve shows a trajectory under another optimistic policy that ensures constraints are satisfied with an ϵ-margin and is appended in the beginning by a small horizon δh to move from $x(k) \to x'$ via policy $\hat{\pi}$. On right, Fig. 1b shows Objective 1, where due to exploration the pessimistic policy set starting from Π_0^{p} expands to $\Pi_{\bar{n}}^{\mathrm{p}}$, and covers the connected true policy set $\Pi_c^{\star,\epsilon}$. Note that $\Pi_c^{\star,\epsilon}$ is a subset of $\Pi^{\star,\epsilon}$, which is, in general, disconnected and thus cannot be discovered by executing only safe policies.

We collect data online as the policy is executed, and construct a growing dataset $\mathcal{D}_n = \{Z, Y\}$ with D_n data points, where $Z = [z_1, \dots, z_{D_n}]$ with $z_k \doteq (x_k, u_k) \in \mathbb{R}^{n_x+n_u}$. The corresponding noisy measurements are $Y = [y_1, \dots, y_{D_n}]$, where each $y_k = \boldsymbol{f}^\star(z_k) + \eta_k$. We use Gaussian process (GP) regression to compute the posterior mean $\mu_{n,i}(z)$ and the covariance $\sigma_{n,i}(z)$ for each $i \in \mathbb{N}_{[1,n_x]}$ using noise standard deviation σ [26, 29]. Here, n denotes the number of model updates, and each update conditions on all the data $\mathcal{D}_n$ collected so far. Based on this, we build a dynamics set

$$\mathcal{F}_n \coloneqq \left\{ \boldsymbol{f} \mid |f_i(z) - \mu_{n',i}(z)| \leq \sqrt{\beta_{n',i}}\sigma_{n',i}(z) \; \forall z \in \mathcal{X} \times \mathcal{U}, n' \in \mathbb{N}_{[0,n]}, i \in \mathbb{N}_{[1,n_x]} \right\}, \tag{4}$$

with a scaling factor $\beta_{n,i} > 0$ and define the confidence width $w_{n,i}(z) \coloneqq 2\sqrt{\beta_{n,i}}\sigma_{n,i}(z)$ quantifying the uncertainty of the dynamics set $\mathcal{F}_n$. We use $w_n(x,u) \coloneqq \max_i w_{n,i}(x,u)$ and $\beta_n \coloneqq \max_i \beta_{n,i}$ to denote the maximum across all state components. The following lemma provides sufficient conditions to obtain well-calibrated bounds $\mathcal{F}_n$ when modeling the unknown dynamics with GPs[1].

Lemma 1 (Well-calibrated model [27, Theorem 3.11]). *Let Assumption 1 hold and* $\sqrt{\beta_{n,i}} \coloneqq B_i + \sqrt{\ln(\det(I_{D_n} + \sigma^{-2} K^i_{D_n})) + 2\ln(n_x/\delta)}$. *Then, it holds that* $\Pr(\boldsymbol{f}^\star \in \mathcal{F}_n, \forall n \in \mathbb{N}) \geq 1 - \delta$.

Assumption 2 (Lipschitz continuity). *The dynamics model* $\boldsymbol{f}^\star$ *is* L_{f}*-Lipschitz,* w_n *is* L'_w*-Lipschitz* $\forall n \in \mathbb{N}$*, the reward* r *is* L'_r*-Lipschitz, and any policy* $\pi \in \Pi_{\mathrm{H}}$ *is* L_π *Lipschitz.*

Under Assumption 1, the dynamics $\boldsymbol{f}^\star$ is Lipschitz continuous if the kernels k^i are Lipschitz continuous, such as the squared exponential or the Matérn kernels [30]. Similarly, Lipschitz continuity of the confidence width w_n can be derived [5]. Moreover, Assumption 2 also implies that closed-loop dynamics $\boldsymbol{f}^\star(x, \pi_h(x))$ is L-Lipschitz continuous with some $L \leq L_{\mathrm{f}}(1 + L_\pi)$ and similarly $w_n(x, \pi(x))$ is L_w-Lipschitz $\forall n \in \mathbb{N}$ and the reward $r(x, \pi(x))$ is L_r-Lipschitz continuous. For short notation we define $L_h \coloneqq \sum_{i=0}^{h-1} L^i$ and $L_{w,h} \coloneqq \sum_{i=0}^{h-1} (L + L_w)^i$.

3 Safe guaranteed dynamics exploration

Main Idea. We execute a policy that pessimistically ensures safety for all dynamics $\boldsymbol{f} \in \mathcal{F}_n$, while optimistically planning to visit informative states with some dynamics $\boldsymbol{f}^s \in \mathcal{F}_n$. If the resulting trajectory of the true system $\boldsymbol{f}^\star$ deviates significantly from the planned trajectory with $\boldsymbol{f}^s$, then we *gain information* about the unknown dynamics by observing this discrepancy. Otherwise, if they are close enough, then we still *gain information* by reaching the intended informative states. In a nutshell; pessimism ensures safety, and no optimistic plan is bad, since we gain information either way.

3.1 Maximum safe dynamics learning via guaranteed exploration in policy space

The exploration process is illustrated in Fig. 1. We start in some (invariant) safe set $\mathbb{X}_n \subseteq \mathcal{X}$, where the dependence on n highlights that the set can also expand with online measurements. We optimize

[1] Notably, our approach equally applies to other probabilistic models, as long as they provide an error bound akin to Lemma 1 for safety and a condition comparable to Assumption 4 to ensure finite-time termination.

a plan that satisfies the constraints and ends again in the safe set for all possible dynamics $\boldsymbol{f} \in \mathcal{F}_n$. In addition, one of the trajectories in this set also (optimistically) reaches an informative state. Intuitively, we achieve *maximum safe dynamics learning* when the set of pessimistically safe policies includes all policies that are safe (with some tolerance ϵ) for the true dynamics. To formalize this, we first define the (unknown) true ϵ-safe policy set:

$$\begin{aligned}\Pi_n^{\star,\epsilon}(X;\mathrm{H}) \coloneqq \{\pi \in \Pi_{\mathrm{H}} \mid \exists x_0 \in X : x_{h+1} = \boldsymbol{f}^{\star}(x_h, u_h) + \eta_h,\ \forall \eta_h \in \mathcal{W},\\ x_h \in \mathcal{X} \ominus \mathcal{B}_{\epsilon}, u_h \coloneqq \pi_h(x_h) \in \mathcal{U} \ominus \mathcal{B}_{\epsilon}, \forall h \in \mathbb{N}_{[0,\mathrm{H}-1]}, x_{\mathrm{H}} \in \mathbb{X}_n \ominus \mathcal{B}_{\epsilon}\}. \quad (5)\end{aligned}$$

This denotes the set of all policies that, when applied to the true system starting from some state in a set $X \subseteq \mathcal{X}$, return the system to the safe set while satisfying all constraints with an ϵ-margin. The user-chosen tolerance ϵ accounts for the fact that dynamics can be learned only up to a nonzero tolerance in finite time. It can be any positive constant, subject to a lower bound proportional to $\tilde{\eta}$, see Appendix B. For example, in the case of no noise $\tilde{\eta} = 0$, the tolerance $\epsilon > 0$ can be chosen arbitrarily small.

To approximate this set via exploration, we define an inner and outer approximation of the true policy set (5), namely, pessimistic and optimistic policy sets, respectively, as follows:

$$\begin{aligned}\Pi_n^{\mathrm{p}}(X;\mathrm{H}) \coloneqq \{\pi \in \Pi_{\mathrm{H}} \mid \exists x_0 \in X : x_{h+1} = \boldsymbol{f}(x_h, u_h) + \eta_h, \forall \eta_h \in \mathcal{W}, \forall \boldsymbol{f} \in \mathcal{F}_n,\\ x_h \in \mathcal{X}, u_h \coloneqq \pi_h(x_h) \in \mathcal{U}, \forall h \in \mathbb{N}_{[0,\mathrm{H}-1]}, x_{\mathrm{H}} \in \mathbb{X}_n\}; \quad (6)\end{aligned}$$

$$\begin{aligned}\Pi_n^{\mathrm{o},\epsilon}(X;\mathrm{H}) \coloneqq \{\pi \in \Pi_{\mathrm{H}} \mid \exists \boldsymbol{f} \in \mathcal{F}_n, \exists x_0 \in X : x_{h+1} = \boldsymbol{f}(x_h, u_h) + \eta_h,\ \forall \eta_h \in \mathcal{W},\\ x_h \in \mathcal{X} \ominus \mathcal{B}_{\epsilon}, u_h \coloneqq \pi_h(x_h) \in \mathcal{U} \ominus \mathcal{B}_{\epsilon}, \forall h \in \mathbb{N}_{[0,\mathrm{H}-1]}, x_{\mathrm{H}} \in \mathbb{X}_n \ominus \mathcal{B}_{\epsilon}\}. \quad (7)\end{aligned}$$

The *pessimistic* policies ensure that *for all dynamics* $\boldsymbol{f} \in \mathcal{F}_n$, the system is safe and returns to the safe set $x_{\mathrm{H}} \in \mathbb{X}_n$, as shown with the dashed region in Fig. 1a. In contrast, the *optimistic* policies ensure *at least one dynamics* is safe while keeping a tolerance $\epsilon > 0$ with respect to the constraint. These set definitions naturally satisfy $\Pi_n^{\mathrm{p}}(X;\mathrm{H}) \subseteq \Pi_n^{\star}(X;\mathrm{H}) \subseteq \Pi_n^{\mathrm{o}}(X;\mathrm{H})$,$\forall n \in \mathbb{N}, X \subseteq \mathcal{X}$, given $\boldsymbol{f}^{\star} \in \mathcal{F}_n$ (cf. Proposition 1 in Appendix B).

Through continuous exploration, we can expand the pessimistic policy set $\Pi_n^{\mathrm{p}}(X;\mathrm{H})$, however, we cannot hope to discover disconnected safe policies which require executing possibly unsafe policies during exploration; see Fig. 1b for an illustration. Hence, we restrict our theoretical analysis to the *connected policy set* $\Pi_c^{\star}(X;\mathrm{H}) \subseteq \Pi^{\star}(X;\mathrm{H})$, which satisfies the following connectedness condition: $\forall \pi^{\star} \in \Pi_c^{\star}(X;\mathrm{H})$, there exists a continuous curve $\rho : [0,1] \to \Pi_c^{\star}(X;\mathrm{H})$, such that $\rho(0) = \pi^{\star}, \rho(1) \in \Pi_n^{\mathrm{p}}(X;\mathrm{H})$. In addition, during recursive planning, the agent may end in some fixed state $x(k) \in \mathbb{X}_n$. But we are interested in the optimistic policy starting from any possible state $x' \in \mathbb{X}_n$, see Fig. 1a. Thus, we need to extend our horizon slightly up to $\Delta\mathrm{H} \in \mathbb{N}$, which allows the policy to steer the system from $x(k) \to x'$ via a policy $\hat{\pi} \in \Pi_{\Delta\mathrm{H}}$ (see controllability property later in Assumption 3). Hence, the best any algorithm can guarantee for exploration is the following:

Objective 1 (Maximum safe dynamics exploration)**.** *There exists a uniform bound $n^{\star} \in \mathbb{N}$, such that after some $\bar{n} \leq n^{\star}$ at the current state $x(k) \in \mathbb{X}_{\bar{n}}$, it holds that:*

$$\forall x' \in \mathbb{X}_{\bar{n}}, \pi^{\star} \in \Pi_{c,\bar{n}}^{\star,\epsilon}(x';\mathrm{H}), \exists \delta h \in \mathbb{N}_{[0,\Delta\mathrm{H}]}, \hat{\pi} \in \Pi_{\delta h} : [\hat{\pi}, \pi^{\star}] \in \Pi_{\bar{n}}^{\mathrm{p}}(x(k);\mathrm{H}+\delta h).$$

Achieving Objective 1 ensures that for every policy $\pi^{\star}$ in the true policy set $\Pi_c^{\star,\epsilon}(\mathbb{X}_{\bar{n}};\mathrm{H})$, there is a corresponding policy in the pessimistic policy set with slightly extended horizon $\mathrm{H}+\delta h$. This implies that after exploration within $\bar{n} \leq n^{\star}$ iterations, the ϵ-true policy set is effectively a subset of the pessimistic policy set. Since we can optimize over the pessimistic policy set, achieving this objective ensures that we have explored enough to recover close-to-optimal policies from the true set; see Section 4.

Remark 1. *We consider two horizons,* H *and* $\mathrm{H}+\delta h$*, where δh considers the time to move in the safe set; see Fig. 1a. This arises since the dynamics in the safe set is unknown. If the dynamics are known inside the safe set, we can exactly steer the system to the desired location in the set $\mathbb{X}_n$, and Objective 1 can be replaced by the more intuitive result:* $\Pi_{\bar{n}}^{\star,\epsilon}(\mathbb{X}_{\bar{n}};\mathrm{H}) \subseteq \Pi_{\bar{n}}^{\mathrm{o},\epsilon}(\mathbb{X}_{\bar{n}};\mathrm{H}) \subseteq \Pi_{\bar{n}}^{\mathrm{p}}(\mathbb{X}_{\bar{n}};\mathrm{H})$.

3.2 Exploration process for maximum safe dynamics learning

Next, we present our novel exploration process, ensuring that, at each planning step, the agent gathers information about the dynamics until a desired tolerance is achieved. Specifically, we define three tolerances: *(i)* ϵ, a user-defined tolerance that guarantees exploration of the policy set (cf. Objective 1), *(ii)* ϵ_d, the tolerance on the learned dynamics, and *(iii)* ϵ_c, the tolerance with

Algorithm 1 Safe guaranteed dynamics exploration

```
1: Initialize: Start at $x(0) \in \mathbb{X}_0$, $\mathcal{F}_0$, $\mathrm{H}_c$, Tol. $\epsilon$, Data $\mathcal{D}_0$
2: for $n = 0, 1, \ldots$ do
3:     $\pi^p \leftarrow$ Solve Problem (8) with current state $x(k)$.
4:     if Problem (8) is infeasible then terminate
5:     $x(k) \leftarrow$ Apply $\pi^p$ to $\boldsymbol{f}^\star$ for $\mathrm{H}_c$ steps and
6:         collect $\mathcal{D}_c := \{(x_{h+1}, z_h) | w_{n-1}(z_h) \geq \epsilon_c\}$.
7:     Update $\mathcal{F}_n$ model with $\mathcal{D}_{n+1} \leftarrow \mathcal{D}_n \cup \mathcal{D}_c$.
```

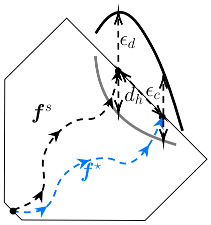


Figure 2: Role of tolerances ϵ_d and ϵ_c in ensuring dynamics exploration.

which measurements are collected. Given a user-specified ϵ (arbitrary subject to noise bound), the tolerances ϵ_d and ϵ_c can be uniquely determined as specified in Appendix B.

Fig. 2 illustrates the relation between ϵ_c and ϵ_d. Suppose that we learn the dynamics up to ϵ_c tolerance. Then, in the worst case, after h-steps, the planned and true path will deviate by at most $(\epsilon_c + \tilde{\eta})L_h$, where the noise bound $\tilde{\eta} \geq \|\eta(k)\| \; \forall \eta \in \mathcal{W}$ and L_h is related to the Lipschitz constant L. Hence, if the distance d_h between the planned and true trajectory after h-steps exceeds $(\epsilon_c + \tilde{\eta})L_h$, then there was at least one point in the trajectory where the dynamics was not known up to ϵ_c tolerance. Hence, by (optimistically) searching for uncertainty $w_n(x_h, u_h)$ greater than $\epsilon_d := \epsilon_c + L_w L_{\mathrm{H}_c}(\epsilon_c + \tilde{\eta})$, we guarantee to learn about the dynamics. Here, $\mathrm{H}_c := \mathrm{H} + \Delta\mathrm{H}$ is a constant horizon with $\Delta\mathrm{H} \geq \delta h$ accounting for the appended trajectory, see Assumption 3 later.

Exploration scheme. Given this intuition, we achieve Objective 1 using the following sampling strategy at the $(n+1)^{th}$ planning iteration:

$$\text{Find } \pi^p \in \Pi_n^{\mathrm{p}}(x(k); \mathrm{H}_c) \; : \; w_n(x_h, u_h) \geq \epsilon_d \text{ for some } \boldsymbol{f} \in \mathcal{F}_n, h \in \mathbb{N}_{[0, \mathrm{H}_c - 1]}, \tag{8}$$

where the state $x_{h+1} = \boldsymbol{f}(x_h, u_h)$ is propagated using action sequence $u_h := \pi_h^p(x_h)$, $h \in \mathbb{N}_{[0,\mathrm{H}_c-1]}$, with the optimized pessimistically safe policy $\pi^p \in \Pi_n^{\mathrm{p}}(x(k); \mathrm{H}_c)$ starting from the current state $x_0 = x(k)$. The optimized policy π^p ensures safe operation for the horizon H_c and ends in the safe set $\mathbb{X}_n$ for all dynamics $\boldsymbol{f} \in \mathcal{F}_n$. In addition, the policy ensures that for at least one dynamics $\boldsymbol{f} \in \mathcal{F}_n$, execution leads to an informative location. Problem (8) provides a sufficient condition to guarantee exploration; Remark 2 outlines objectives that accelerate complete exploration, and we further expand it to maximize cumulative rewards in Section 4. The finite-horizon Problem (8) can be solved efficiently using optimization algorithms tailored for MPC, see Remark 3 for details on the implementation.

Safe dynamics exploration process. Algorithm 1 summarizes the proposed framework. The agent starts at location $x(0) \in \mathbb{X}_0$ and computes the pessimistic policy by solving Problem (8) in Line 3. The agent applies the pessimistic policy to the true (unknown) dynamics, yielding a state sequence $x_h, h \in \mathbb{N}_{[0,\mathrm{H}_c]}$ and collects data along the way, which satisfy $w_{n-1}(z_h) \geq \epsilon_c$ in Line 6. As our theoretical analysis will reveal, there is always at least one such data point, given that the policy is a solution of Problem (8). We update the dynamical model with the collected data, and the agent ends up at some state $x(k) = x_{\mathrm{H}_c} \in \mathbb{X}_n$. We then resolve Problem (8) to identify the next safe exploration policy and the process continues until Problem (8) is infeasible, i.e., $w_n(x_h, u_h) < \epsilon_d$ for any state and action x_h, u_h that can be reached with any dynamics $\boldsymbol{f} \in \mathcal{F}_n$ using any pessimistically safe policy $\Pi_n^{\mathrm{p}}(x(k); \mathrm{H}_c)$. This implies that there are no more safely reachable informative states and thus the algorithm terminates in Line 4.

Remark 2. *Any policy that satisfies the constraints of Problem* (8) *can be applied and will ensure satisfaction of Objective 1. A particularly attractive objective is* $\max_{\pi \in \Pi_n^{\mathrm{p}}(x_s; \mathrm{H}_c)} \max_{\boldsymbol{f} \in \mathcal{F}_n} \sum_{h=0}^{\mathrm{H}_c - 1} w_n(x_h, u_h)$*, which ensures that the constraints of* (8) *hold if the reward exceeds* $\epsilon_d \mathrm{H}_c$*. If rewards are less than* ϵ_d*, Problem* (8) *is infeasible, which ensures termination of Algorithm 1.*

3.3 Theoretical guarantees

In the following, we prove that the sampling rule (8) guarantees *maximum safe dynamics exploration* (Objective 1). To do so, we begin by formalizing the safe set assumption as follows:

Assumption 3 (Safe set)**.** *The agent starts in a known safe set* $x(0) \in \mathbb{X}_0$ *which* $\forall n \geq 0$ *satisfies*

- *Monotonicity:* $\mathbb{X}_n \subseteq \mathbb{X}_{n+1}$*;*
- *Invariance:* $\mathbb{X}_n$ *is a robust positive invariant set with a terminal policy* $\pi_{\mathrm{f}} \in \Pi$*, i.e.,* $\forall \boldsymbol{f} \in \mathcal{F}_n$*,* $\eta \in \mathcal{W}$*,* $x \in \mathbb{X}_n$*,* $\boldsymbol{f}(x, \pi_{\mathrm{f}}(x)) + \eta \in \mathbb{X}_n \ominus \mathcal{B}_\epsilon, x \in \mathcal{X} \ominus \mathcal{B}_\epsilon, \pi_{\mathrm{f}}(x) \in \mathcal{U} \ominus \mathcal{B}_\epsilon$*;*

- *Controllability: All dynamics in the set $\mathcal{F}_n$ can be controlled to any state in the safe set while satisfying constraints within time $\Delta\mathrm{H}$, i.e., $\forall x_s, x_e \in \mathbb{X}_n$, $\boldsymbol{f} \in \mathcal{F}_n$, $\exists \delta h \in \mathbb{N}_{[0,\Delta\mathrm{H}]}, \hat{\pi} \in \Pi_{\delta h}$: $x_0 = x_s, x_{h+1} = \boldsymbol{f}(x_h, u_h), x_h \in \mathbb{X}_n \ominus \mathcal{B}_\epsilon, u_h := \hat{\pi}_h(x_h) \in \mathcal{U} \ominus \mathcal{B}_\epsilon, \forall h \in \mathbb{N}_{[0,\delta h-1]}, x_{\delta h} = x_e$.*

The assumption is comparable to the safe initial seed for safe exploration [12, 17, 18, 31] and safe terminal sets assumed in the MPC literature [32–35]. The safe set can be designed by sampling the dynamics $\boldsymbol{f} \in \mathcal{F}_n$ and computing a common Lyapunov function around the equilibrium point, similar to [22, Sec. 6.1]. The safe set $\mathbb{X}_n$ can be expanded with n as the agent gathers more data about the dynamics and shrinks the dynamics set $\mathcal{F}_n$. To show finite time termination of this simple sampling strategy, we also consider a standard regularity assumption on the kernel. For this, we define maximum information capacity $\gamma_n := \sup_{\mathcal{D}_n \subseteq \mathcal{X} \times \mathcal{U} : |\mathcal{D}_n| \leq D_0 + n\mathrm{H}_c} I(Y_{\mathcal{D}_n}; \boldsymbol{f}^\star_{\mathcal{D}_n})$ [28] that upper bounds the information that can be obtained with a finite number of measurements D_n collected at any set $\mathcal{D}_n$, where I denotes the mutual information associated to the GP model, see Lemma 6 in Appendix B for details.

Assumption 4. *$\beta_n\gamma_n$ grows sublinearly in n, i.e., $\beta_n\gamma_n < \mathcal{O}(n)$.*

Such an assumption is common in most prior works [12, 14–16, 28, 36, 37] to establish sample complexity or sublinear regret results. Indeed, due to the bounded B_i (Assumption 1), $\beta_n\gamma_n$ grows sublinear in n for compact domains $\mathcal{X}$ and commonly used kernels, e.g., linear, squared exponential, Matérn, etc., with sufficient eigen decay[28, 38]. The following theorem guarantees that Algorithm 1 ensures maximum exploration and safety of dynamics for the true (unknown) system (1).

Theorem 1. *Let Assumptions 1 to 4 hold. Let $n^\star$ be the largest integer satisfying $\frac{n^\star}{\beta_{n^\star}\gamma_{n^\star}} \leq \frac{C_1}{\epsilon_c^2}$ with $C_1 = 8\mathrm{H}_c / \log(1 + \mathrm{H}_c\sigma^{-2})$. Then, with at least $1 - \delta$ probability, Algorithm 1 guarantees*

- *safety for all times: $x(k) \in \mathcal{X}, u(k) \in \mathcal{U}\ \forall k \in \mathbb{N}$;*
- *termination after $\bar{n} \leq n^\star$ iterations;*
- *Objective 1, ensuring maximum safe dynamics exploration.*

Thus, by executing a pessimistically safe policy and collecting data where uncertainty is larger than ϵ_c via (8), we achieve dynamics exploration up to the ϵ_d tolerance while being safe at all times for uncertain nonlinear systems. This corresponds to maximal safe exploration of the policies up to arbitrary tolerance ϵ subject to noise (Objective 1), with the total exploration time bounded by $k \leq n^\star\mathrm{H}_c$. The complete proof is provided in Appendix B.

4 Reward maximization with intrinsic exploration

Building on the general framework for safe dynamics exploration from Section 3, we propose SAGEDYNX, an algorithm that focuses on maximizing rewards by reducing uncertainty *only* in the regions essential to achieve optimal operation. SAGEDYNX replans directly whenever new measurements are collected instead of returning back to the safe set, thus enhancing efficiency. Furthermore, the algorithm explores in task-oriented fashion and learns a safe policy π^{r} that achieves the following objective.

Objective 2 (Safe reward maximization)**.** *There exists $\bar{n} \leq n^\star$ and a constant $K > 0$, such that the known safe policy $\pi^{\mathrm{r}} \in \Pi^\star_{\bar{n}}(x(k), \mathrm{H} + \delta h)$ computed at the current state $x(k) \in \mathbb{X}_n$ satisfies:*

$$J(x(k), \boldsymbol{f}^\star; \pi^{\mathrm{r}}) \geq \max_{x^\star \in \mathbb{X}_{\bar{n}}, \pi^\star \in \Pi^{\star;\epsilon}_{c,\bar{n}}(\mathbb{X}_{\bar{n}};\mathrm{H})} J(x^\star, \boldsymbol{f}^\star; \pi^\star) - K\epsilon. \tag{9}$$

Objective 2 ensures that, despite a-priori unknown dynamics, we obtain a policy with a performance that is close to the optimal policy $\pi^\star$ starting from the best state $x^\star$ within the safe set. To achieve Objective 2, naively, one could explore the complete set of pessimistic policies using Algorithm 1 and then maximize $J(x, \pi^{\mathrm{p}}; f^\star)$ among the known safe policies, yielding a sequential two-stage algorithm. However, exploring the complete pessimistic policy set without considering the objective would be extremely inefficient. Therefore, we propose an exploration strategy that leverages the optimistic policy set, which excludes only those policies that are provably unsafe for all dynamics. Hence, reward maximization in the optimistic set guarantees maximizing among the plausible policies. To formalize this, we define the optimistic and pessimistic problems, respectively, as follows:

$$J(x^o, \boldsymbol{f}^o; \pi^{\mathrm{o}}) := \max_{x \in \mathbb{X}_n, \boldsymbol{f} \in \mathcal{F}_n, \pi \in \Pi^{\mathrm{o},\epsilon}_n(x;\mathrm{H})} J(x, \boldsymbol{f}; \pi), \quad J^{\mathrm{p}}(x^{\mathrm{p}}, \boldsymbol{\mu}_n; \pi^{\mathrm{p}}) := \max_{x \in \mathbb{X}_n, \pi \in \Pi^{\mathrm{p},\epsilon'}_n(x;\mathrm{H})} J^{\mathrm{p}}(x, \boldsymbol{\mu}_n; \pi), \tag{10}$$

$$\text{where, } J^{\mathrm{p}}(x, \boldsymbol{\mu}_n; \pi) = J(x, \boldsymbol{\mu}_n; \pi) - \mathbb{E}\left[L_r \sum\nolimits_{h=0}^{\mathrm{H}-1} \sum\nolimits_{i=0}^{h} L^i w_n(x_i, \pi_i(x_i)) | x_0 = x, \pi\right], \tag{11}$$

Algorithm 2 Reward maximization with intrinsic exploration (SAGEDYNX)

```
1:  Initialize: Start at $x(0) \in \mathbb{X}_0$, $\mathcal{F}_0$, $\mathrm{H}_c$, Tolerance $\epsilon$, Data $\mathcal{D}_0$.
2:  for $n = 0, 1, \ldots$ do
3:      $J(x^o, \boldsymbol{f}^o; \pi^{\mathrm{o}}) \leftarrow$ Solve the optimistic problem (10) (left).
4:      $J^{\mathrm{p}}(x^{\mathrm{p}}, \boldsymbol{\mu}_n; \pi^{\mathrm{p}}) \leftarrow$ Solve the pessimistic problem (10) (right).
5:      if $J^{\mathrm{p}}(x^{\mathrm{p}}, \boldsymbol{\mu}_n; \pi^{\mathrm{p}}) \geq J(x^o, \boldsymbol{f}^o; \pi^{\mathrm{o}}) - \epsilon K$ then
6:          $x(k) \leftarrow$ Return to the safe set using the previously optimized policy.
7:          Return $\pi^{\mathrm{r}} \leftarrow [\hat{\pi}, \pi^{\mathrm{p}}]$,where $\hat{\pi}$ steers $x(k)$ to $x^{\mathrm{p}}$ with $\boldsymbol{\mu}_n$ and terminate.
8:      $\pi_e^{\mathrm{p}}, \nu \leftarrow$ Solve Problem (12).
9:      if $\nu = 0$ then                                  (Task oriented safe expansion)
10:         $x(k) \leftarrow$ Execute $\pi_e^{\mathrm{p}}$ on $\boldsymbol{f}^\star$ until collect $|\mathcal{D}_c| \geq 1$ data,$\mathcal{D}_c{:=}\{(x_{h+1}, z_h)|w_{n-1}(z_h) \geq \epsilon_c\}$.
11:     else                    (No informative location found, Return to the safe set)
12:         $x(k) \leftarrow$ Continue with policy optimized at $n-1$ until $\mathbb{X}_n$ is reached.
13:         $\pi_e^{\mathrm{p}} \leftarrow$ Solve Problem (12) with $\nu = 0$ (hard constraint).
14:         $x(k) \leftarrow$ Execute $\pi_e^{\mathrm{p}}$ until collect $|\mathcal{D}_c| \geq 1$ data.
15:     Update $\mathcal{F}_n$ with $\mathcal{D}_{n+1} \leftarrow \mathcal{D}_n \cup \mathcal{D}_c$.          (Update model)
```

and $x^o, \boldsymbol{f}^o, \pi^{\mathrm{o}}, x^{\mathrm{p}}, \pi^{\mathrm{p}}$ denote the optimized variables. The pessimistic problem maximizes J^{p}, which provides a pessimistic lower bound on the reward J for all $\boldsymbol{f} {\in} \mathcal{F}_n$. The states x_i are propagated with noise under the mean dynamics $\boldsymbol{\mu}_n$, which we use for computational simplicity, although J^{p} can, in principle, be defined using any dynamics. Here, $\epsilon' \in (0, \epsilon)$ is a small tolerance to account for the different initial conditions in the safe set $\mathbb{X}_n$ using a controllability argument (see Appendix C later). Since the pessimistic problem provides a lower bound on the objective under the true dynamics, using it for termination allows early stopping, that is, without accurately knowing the dynamics everywhere, it still lets us recover close to optimal performance.

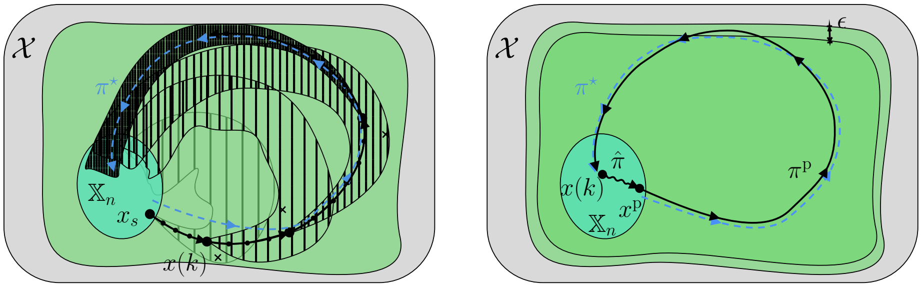
Reward maximization algorithm–SAGEDYNX. The method is summarized in Algorithm 2. Overall, we optimistically maximize the rewards while pessimistically ensuring safety. We start by solving the optimistic problem in Line 3, optimizing over the initial state x^o, optimistic dynamics, resulting in the policy that could potentially achieve the maximum rewards. Then we solve the pessimistic problem in Line 4 that ensures the safety of the unknown system while maximizing the rewards. Suppose that the pessimistic rewards are sufficiently close to the optimistic rewards. In that case, the algorithm terminates in Line 5, signifying that the agent has explored enough to converge close to the optimal solution. Otherwise, to continue exploring, we solve the following problem in Line 8:

$$\pi_e^{\mathrm{p}} = \underset{\nu, \pi \in \Pi_n^{\mathrm{p}}(x(k); \mathrm{H}_c)}{\arg\max} \; -\lambda \nu + J^{\mathrm{any}}(x(k), n; \pi), \text{ s.t. } w_n(x_h, u_h) \geq \epsilon_d - \nu, \nu \geq 0, h {\in} \mathbb{N}_{[0, \mathrm{H}_c - 1]}, \quad (12)$$

where $\lambda > 0$ is a user-defined large penalty and ν is a slack variable to ensure the feasibility of the problem. Here, x_h is the propagated state using any $\boldsymbol{f} \in \mathcal{F}_n$ with optimized policy π. J^{any} can be freely designed and can also depend on $\mathcal{F}_n, \pi^{\mathrm{o}}$ or $\boldsymbol{f}^o$ indicated by n in (12); see Remark 4 for objectives that may speed up exploration by using the solution of the optimistic problem.

In Problem (12), with $\nu = 0$ we are guaranteed to have sufficient information of $w_n(x_h, u_h) \geq \epsilon_d$ while ensuring a *task-oriented* approach by maximizing the J^{any} objective. We execute the resulting policy π_e^{p} until some $|\mathcal{D}_c| \geq 1$ measurements are collected and update the dynamics model using the gathered data. Then we replan directly at the state where we collected the measurement, without first going back to the safe set, as shown in Fig. 3a. If $\nu > 0$, it implies that the agent has not found an informative location from the current state. In this case, it returns to the safe set, and then solves Problem (12) (with $\nu = 0$, hard constraint) to explore another region. This hard constraint problem is guaranteed to be feasible (c.f. Theorem 2) and yields another informative state $w_n(x, u) > \epsilon_d$, as otherwise the algorithm would already have terminated in Line 5. Notably, the case $\nu > 0$ (Line 13) is required only for technical reasons, but typically does not occur in practice. The algorithm mainly revolves around Line 8, optimizes the objective, collects data, replans, and keeps on going until Objective 2 can be achieved, see Fig. 3. All of this is achieved while ensuring safety and maintaining a safe returnable path (pessimistic policy to the safe set $\mathbb{X}_n$) without actually returning to the safe set.

Remark 3 (Implementation)**.** *Problems* (8) *and* (12) *require computing policies $\pi \in \Pi_n^{\mathrm{p}}$ that ensure constraint satisfaction for all dynamics $f \in \mathcal{F}_n$ over a finite horizon* H_c*. This can be implemented efficiently by over-approximating this reachable set using a finite number of samples from $f \sim \mathcal{GP}$,*



(a) Re-planning after collecting measurements

(b) Returned policy behaves close to optimal

Figure 3: Illustration of SAGEDYNX algorithm during a) exploration and b) convergence after satisfying the termination criteria. The cyan region is the safe set $\mathbb{X}_n$, the green region is the constraints $\mathcal{X}$, and the blue dashed line shows the clairvoyant optimal trajectory within $\mathbb{X}_n$ satisfying constraints with margin ϵ. In Fig. 3a, the shaded region shows the reachable set under the optimized pessimistic policy by (12), which starts at the current state $x(k)$, ensures all the dynamics satisfy the constraints, and returns to the safe set. The black line shows the agent's executed trajectory, with small dots marking collected data and large dots indicating the time of model updates. With every model update (increasing n), the reachable set (represented by darker shades) predicted with a given policy π shrinks since the model uncertainty reduces with data. The agent keeps on replanning while ensuring returnability to a known safe set, but without having to actually return. Once the termination criteria is satisfied, the agent returns to the safe set. As shown in Fig. 3b, it then executes the returned policy which first navigates in the safe set for small δh horizon (wiggly line), and then executes the optimized policy π^{p} shown by black line which closely matches the optimal trajectory (blue dashed line).

as shown in [22]. By restricting the policy parameterization to simple affine policies, the resulting optimization problem can be solved using standard optimization algorithms for finite-horizon optimal control, while maintaining the safety guarantees.

Theoretical guarantees. This section presents the theoretical results for the proposed algorithm SAGEDYNX. To allow for earlier termination and be able to recover the close-to-optimal policy from any state in the safe set $\mathbb{X}_n$, we make the following assumption on the dynamics in the safe set.

Assumption 5 (Dynamics in safe set)**.** $\boldsymbol{f}^\star$ *is known up to* $\epsilon_c \geq 0$ *tolerance in the safe set.*

This implies $w_{n-1}(x,u) < \epsilon_c,\ \forall x \in \mathbb{X}_n,\ u \in \mathcal{U},\ n \in \mathbb{N}$. The following theorem guarantees that the Algorithm 2 provides a close-to-optimal policy while being safe for the unknown system (1).

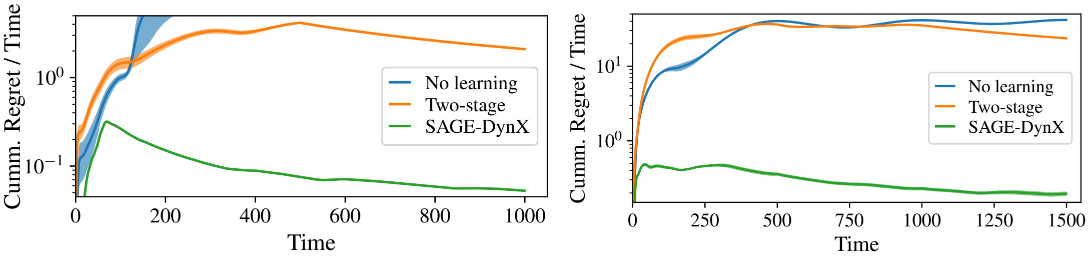
Theorem 2. *Let Assumptions 1 to 5 hold. Consider* $n^\star$ *as in Theorem 1. With probability atleast* $1-\delta$*,*

- *All the optimization problems in Algorithm 2 are feasible for* $\forall n \geq 0$*;*
- *Algorithm 2 guarantees safety for the unknown system at all times:* $x(k) \in \mathcal{X}, u(k) \in \mathcal{U}, \forall k \in \mathbb{N}$*;*
- *Algorithm 2 is guaranteed to terminate in at most* $n^\star$ *iterations;*
- *Algorithm 2 returns policy* π^{r} *that satisfies Objective 2 i.e., achieves close-to-optimal performance.*

Thus, Algorithm 2 is guaranteed to terminate in a finite time and yields a policy that achieves a performance that is arbitrarily close to optimal performance, i.e., achieves Objective 2. Note that once the termination criterion is satisfied, the agent can execute the policy π^{p} and does not need to explore the dynamics everywhere. Since exploration is essential to guarantee optimality, in the worst case, the algorithm may need to learn dynamics everywhere, and the exploration time for this scenario is bounded by $k \leq n^\star \mathrm{H}_c$. The detailed proof can be found in Appendix C.

5 Experiments

We empirically evaluate the performance of SAGEDYNX in multiple environments. Here we present experiments on two representative tasks: car racing and drone navigation under aerodynamic effects. We compare SAGEDYNX against two baselines: i) No learning, which attempts to solve the task using only prior data without gathering new information (measurements), ii) Two-stage, which first explores the dynamics by gathering online data (similar to Algorithm 1) and then optimizes the objective with the learned dynamics. This baseline is akin to [17], which pessimistically explores and then optimizes, although the considered baseline requires no resets. Additionally, we compare performance against a clairvoyant agent that has exact knowledge of the dynamics, representing the optimal performance.



(a) Drone navigation

(b) Car Racing

Figure 4: Cumulative regret over time (averaged across runs) in different environments. SAGEDYNX achieves an order of magnitude lower regret compared to the baselines in both experiments.

We implement the optimization problems as described in Remark 3 with a sampling-based approach. The implementation is based on `Python` interfaces of `acados` [39] and `CasADi` [40]; for efficient GP sampling, we employ `GPyTorch` [41]. For each environment, we report the average with a confidence bound of 1 standard deviation computed on 10 different random seeds. The computation of control input is fast (real-time feasible), for instance, on an i7-11800H processor (2.30GHz) with RTX A2000, the computation of a single control sequence took 32.0 ± 7.7 ms for the car environment. While this section highlights the core experiments, additional results and implementation details are provided in Appendix D. Below we explain our main observations for each environment:

Drone navigation. The dynamics model is a 6-dimensional nonlinear system, representing position, angular orientation, linear velocities, and angular rates from [42, 43]. The control inputs are two-dimensional, corresponding to the thrust force produced by the propellers. The dynamics also include disturbances corresponding to aerodynamic effects. The task requires the drone to track a heart-shaped trajectory while satisfying box constraints in the state space. The rewards correspond to the accuracy of tracking a time-varying heart-shaped reference trajectory. Fig. 4a shows the performance of SAGEDYNX compared to the two baselines. We plot cumulative regret over time, where regret is computed with the position difference from the clairvoyant system at any time step. The no-learning baseline, due to high model uncertainty, quickly fails to maintain accurate tracking. For the two stage baseline, we explore dynamics for the first 500 time steps and then maximize the rewards. We observe a high initial regret due to random exploration of the dynamics, and then it starts to decay after 500 steps. In contrast, SAGEDYNX initially deviates in the beginning to gather data and then quickly comes closer to the optimal behaviour, as evident by the steady decline in the regret shown in Fig. 4a.

Car racing is an interesting application in which, while racing, the model can be learned online. The car dynamics model is given by a 4D nonlinear system representing position, orientation, and velocity from [44]. The control commands are two-dimensional, representing throttle and steering. The task requires the car to follow a reference path while satisfying the track constraints. Similarly to the drone task, the car tries to track a reference, and the rewards are assigned as per the accuracy of tracking. Fig. 4b shows cumulative regret over time. Both the no-learning and two-stage algorithms have high regret due to uncertain dynamics and random exploration, respectively. After 1000 steps, the two-stage algorithm tries to maximize reward, and thus, we see a decrease in the regret. In contrast to baselines, SAGEDYNX attains an order of magnitude lower regret. Initially, SAGEDYNX's regret increases attributed to exploring the dynamics, and then comes closer to the optimal behaviour as shown in Fig. 4b.

6 Conclusion

We proposed a general framework for safe, non-episodic online learning of unknown dynamics, introducing the notion of guaranteed exploration in policy space. To our knowledge, this is the first theoretical analysis that successfully addresses this problem. Building on this, we proposed a theoretically grounded and efficient reward-maximization algorithm that achieves near-optimal performance while learning the dynamics only where necessary for optimality. The proposed approach has two key limitations: a) We utilize a robust bound on the noise $\eta(k) \in \mathcal{W}$, which may be conservative. This can be naturally addressed by replacing the high-probability safety guarantees with (weaker) probabilistic safety guarantees using standard stochastic prediction methods [45, 46]. b) Although we ensure safety for all times $k \in \mathbb{N}$, our performance bound is only valid for rewards over a finite horizon H. Providing similar performance bounds for infinite-horizon performance would require more involved tools, such as (stochastic) turnpikes [47].

References

[1] Cuebong Wong, Erfu Yang, Xiu-Tian Yan, and Dongbing Gu. Autonomous robots for harsh environments: a holistic overview of current solutions and ongoing challenges. *Systems Science & Control Engineering*, 6(1):213–219, 2018.

[2] Juraj Kabzan, Miguel I Valls, Victor JF Reijgwart, Hubertus FC Hendrikx, Claas Ehmke, Manish Prajapat, Andreas Bühler, Nikhil Gosala, Mehak Gupta, Ramya Sivanesan, et al. Amz driverless: The full autonomous racing system. *Journal of Field Robotics*, 37(7):1267–1294, 2020.

[3] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018.

[4] Sham Kakade, Akshay Krishnamurthy, Kendall Lowrey, Motoya Ohnishi, and Wen Sun. Information theoretic regret bounds for online nonlinear control. *Advances in Neural Information Processing Systems*, 33:15312–15325, 2020.

[5] Sebastian Curi, Felix Berkenkamp, and Andreas Krause. Efficient model-based reinforcement learning through optimistic policy search and planning. *Advances in Neural Information Processing Systems*, 33:14156–14170, 2020.

[6] Eitan Altman. *Constrained Markov decision processes*. Routledge, 2021.

[7] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pages 22–31. PMLR, 2017.

[8] Dongsheng Ding, Kaiqing Zhang, Tamer Basar, and Mihailo Jovanovic. Natural policy gradient primal-dual method for constrained markov decision processes. *Advances in Neural Information Processing Systems*, 33:8378–8390, 2020.

[9] Yarden As, Ilnura Usmanova, Sebastian Curi, and Andreas Krause. Constrained policy optimization via bayesian world models. *arXiv preprint arXiv:2201.09802*, 2022.

[10] Adrian Müller, Pragnya Alatur, Volkan Cevher, Giorgia Ramponi, and Niao He. Truly no-regret learning in constrained mdps. *arXiv preprint arXiv:2402.15776*, 2024.

[11] Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, and Alois Knoll. A review of safe reinforcement learning: Methods, theories and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[12] 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.

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

[14] 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.

[15] Manish Prajapat, Matteo Turchetta, Melanie Zeilinger, and Andreas Krause. Near-optimal multi-agent learning for safe coverage control. *Advances in Neural Information Processing Systems*, 35:14998–15012, 2022.

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

[17] Yarden As, Bhavya Sukhija, Lenart Treven, Carmelo Sferrazza, Stelian Coros, and Andreas Krause. Actsafe: Active exploration with safety constraints for reinforcement learning. *arXiv preprint arXiv:2410.09486*, 2024.

[18] Manish Prajapat, Johannes Köhler, Matteo Turchetta, Andreas Krause, and Melanie N Zeilinger. Safe guaranteed exploration for non-linear systems. *IEEE Transactions on Automatic Control*, 2025.

[19] 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*, 3:269–296, 2020.

[20] Lukas Brunke, Melissa Greeff, Adam W Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5, 2021.

[21] Torsten Koller, Felix Berkenkamp, Matteo Turchetta, and Andreas Krause. Learning-Based Model Predictive Control for Safe Exploration. In *Proc. IEEE Conference on Decision and Control (CDC)*, 2018.

[22] Manish Prajapat, Johannes Köhler, Amon Lahr, Andreas Krause, and Melanie N. Zeilinger. Finite-sample-based reachability for safe control with gaussian process dynamics. *arXiv preprint arXiv:2505.07594*, 2025.

[23] Tor Aksel N Heirung, Joel A Paulson, Shinje Lee, and Ali Mesbah. Model predictive control with active learning under model uncertainty: Why, when, and how. *AIChE Journal*, 64(8):3071–3081, 2018.

[24] 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.

[25] Raffaele Soloperto, Matthias A. Müller, and Frank Allgöwer. Guaranteed closed-loop learning in model predictive control. *IEEE Transactions on Automatic Control*, 68(2):991–1006, 2023.

[26] Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning*. PMLR, 2017.

[27] Yasin Abbasi-Yadkori. *Online learning for linearly parametrized control problems*. PhD thesis, University of Alberta, 2013.

[28] Niranjan Srinivas, Andreas Krause, et al. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE transactions on information theory*, 58(5):3250–3265, 2012.

[29] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

[30] Christian Fiedler. Lipschitz and hölder continuity in reproducing kernel hilbert spaces. *arXiv preprint arXiv:2310.18078*, 2023.

[31] 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.

[32] Kim P Wabersich et al. Data-driven safety filters: Hamilton-jacobi reachability, control barrier functions, and predictive methods for uncertain systems. *IEEE Control Systems Magazine*, 43(5), 2023.

[33] Danilo Saccani, Leonardo Cecchin, and Lorenzo Fagiano. Multitrajectory model predictive control for safe uav navigation in an unknown environment. *IEEE Transactions on Control Systems Technology*, 2022.

[34] Raffaele Soloperto, Ali Mesbah, and Frank Allgöwer. Safe exploration and escape local minima with model predictive control under partially unknown constraints. *IEEE Transactions on Automatic Control*, 2023.

[35] Johannes Köhler, Matthas A Müller, and Frank Allgöwer. Analysis and design of model predictive control frameworks for dynamic operation–an overview. *Annual Reviews in Control*, 57:100929, 2024.

[36] 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.

[37] Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning*. PMLR, 2017.

[38] 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.

[39] Robin Verschueren, Gianluca Frison, Dimitris Kouzoupis, Jonathan Frey, Niels van Duijkeren, Andrea Zanelli, Branimir Novoselnik, Thivaharan Albin, Rien Quirynen, and Moritz Diehl. Acados—a modular open-source framework for fast embedded optimal control. *Math. Prog. Comp.*, 14(1), 2022.

[40] Joel A. E. Andersson, Joris Gillis, Greg Horn, James B. Rawlings, and Moritz Diehl. CasADi: A software framework for nonlinear optimization and optimal control. *Math. Prog. Comp.*, 11(1), 2019.

[41] Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[42] Sumeet Singh, Benoit Landry, Anirudha Majumdar, Jean-Jacques Slotine, and Marco Pavone. Robust feedback motion planning via contraction theory. *The International Journal of Robotics Research*, 42(9):655–688, 2023.

[43] András Sasfi, Melanie N Zeilinger, and Johannes Köhler. Robust adaptive MPC using control contraction metrics. *Automatica*, 155:111169, 2023.

[44] Jason Kong, Mark Pfeiffer, Georg Schildbach, and Francesco Borrelli. Kinematic and dynamic vehicle models for autonomous driving control design. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 1094–1099, 2015.

[45] Ali Mesbah. Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems Magazine*, 36(6):30–44, 2016.

[46] Yunke Ao, Johannes Köhler, Manish Prajapat, Yarden As, Melanie Zeilinger, Philipp Fürnstahl, and Andreas Krause. Stochastic model predictive control for sub-gaussian noise. *arXiv preprint arXiv:2503.08795*, 2025.

[47] Jonas Schießl, Michael H Baumann, Timm Faulwasser, and Lars Grüne. On the relationship between stochastic turnpike and dissipativity notions. *IEEE Transactions on Automatic Control*, 2024.

[48] Kendall Lowrey, Aravind Rajeswaran, Sham Kakade, Emanuel Todorov, and Igor Mordatch. Plan online, learn offline: Efficient learning and exploration via model-based control. *arXiv preprint arXiv:1811.01848*, 2018.

[49] Weiye Zhao, Rui Chen, Yifan Sun, Tianhao Wei, and Changliu Liu. State-wise constrained policy optimization. *arXiv preprint arXiv:2306.12594*, 2023.

[50] Moritz A Zanger, Karam Daaboul, and J Marius Zöllner. Safe continuous control with constrained model-based policy optimization. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3512–3519. IEEE, 2021.

[51] Yonathan Efroni, Shie Mannor, and Matteo Pirotta. Exploration-exploitation in constrained mdps. *arXiv preprint arXiv:2003.02189*, 2020.

[52] Ian Osband, Daniel Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. *Advances in Neural Information Processing Systems*, 26, 2013.

[53] Zdravko I. Botev, Dirk P. Kroese, Reuven Y. Rubinstein, and Pierre L'Ecuyer. Chapter 3 - the cross-entropy method for optimization. In C.R. Rao and Venu Govindaraju, editors, *Handbook of Statistics*, volume 31 of *Handbook of Statistics*, pages 35–59. Elsevier, 2013.

[54] Thomas M Moerland, Joost Broekens, Aske Plaat, Catholijn M Jonker, et al. Model-based reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 16(1):1–118, 2023.

[55] Aaron D Ames, Xiangru Xu, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2016.

[56] Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[57] Abdullah Tokmak, Kiran G Krishnan, Thomas B Schön, and Dominik Baumann. Safe exploration in reproducing kernel hilbert spaces. *arXiv preprint arXiv:2503.10352*, 2025.

[58] Albert Gassol Puigjaner, Manish Prajapat, Andrea Carron, Andreas Krause, and Melanie N Zeilinger. Performance-driven constrained optimal auto-tuner for MPC. *IEEE Robotics and Automation Letters*, 2025.

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

[60] James Blake Rawlings, David Q Mayne, and Moritz Diehl. *Model predictive control: theory, computation, and design*, volume 2. Nob Hill Publishing Madison, WI, 2017.

[61] Boris Houska and Mario E Villanueva. Robust optimization for mpc. *Handbook of model predictive control*, pages 413–443, 2019.

[62] Ali Mesbah. Stochastic model predictive control with active uncertainty learning: A survey on dual control. *Annual Reviews in Control*, 45:107–117, 2018.

[63] Manish Prajapat, Amon Lahr, Johannes Köhler, Andreas Krause, and Melanie N. Zeilinger. Towards safe and tractable gaussian process-based mpc: Efficient sampling within a sequential quadratic programming framework. In *Proc. IEEE 63rd Conference on Decision and Control (CDC)*, pages 7458–7465, 2024.

A Extended related works

	Safety (during learning)	Safety (final policy)	Optimality	Non-episodic (works without resets)	Handle unknown dynamics
RL [5]	✗	✗	✓	✗	✓
CMDP [51]	✗	(✓)	✓	✗	✓
Safe exploration CMDP [17]	(✓)	(✓)	✓	✗	✓
Safety Augmentation [32]	✓	✓	✗	✓	✓
Safe constraint exploration [18]	✓	✓	(✓)	✓	✗
SageDynX (Ours)	✓	✓	✓	✓	✓

Table 1: Comparison of different approaches on safety, optimality, non-episodic, and handling unknown dynamics. In the table above, the bracket symbol (✓) denotes that the property holds but in a different form, e.g., safety holds in expectation for CMDPs. As shown in the table above, SAGEDYNX is the only algorithm that has all the properties, i.e., it guarantees optimality and safety during the entire execution while operating in a non-episodic setting with unknown dynamics.

Model-based RL and constrained MDPs. Model-based reinforcement learning (MBRL) approaches first learn a predictive model of the environment's dynamics, which is then leveraged for decision-making (either control, planning, or policy optimization). To effectively learn dynamics, MBRL methods employ various exploration strategies, for example, optimism-based [4, 5], posterior (Thompson) sampling [52], and deep ensemble-based [3], among others. Once the model is updated, a range of planning and policy optimization strategies can be applied to maximize cumulative rewards, such as the standard policy gradient [5], as well as sampling-based planners like cross-entropy method [53], MPPI [4]. These approaches can be combined with receding horizon control and re-planning techniques [48], for a general overview, see [54]. All these approaches are designed for unconstrained environments and do not account for state or action constraints.

Particularly appealing regret bounds have recently been shown in [4], which exhibit a polynomial dependence on the planning horizon H. In contrast, our analysis yields an exponential dependence on the horizon length when the Lipschitz constant $L > 1$. This is due to different assumptions: we assume Lipschitz continuity, whereas [4] assumes Gaussian noise and a bounded second moment of cost under the worst policy, for an unbounded (unconstrained) domain. To clarify the differences, consider the linear quadratic regulator (LQR) setting, where (see [4, Remark 3.6]) a stabilizing policy is required to satisfy the bounded second moment of cost assumption. This sufficient condition corresponds to the (weighted [22, Remark 2]) Lipschitz constant $L < 1$, under which our analysis also shows a similar polynomial dependence on the horizon.

A more closely related line of work involves constrained Markov decision processes (CMDPs) [6], which enforce safety through constraints on the expected cumulative cost [7, 8]. Due to the similarity in structure between cumulative rewards and costs, these approaches often leverage policy gradient methods, typically solved using Lagrangian multipliers or primal-dual techniques. In contrast, our work has state-wise (hard) constraints [13, 49], which are more strict; that is, a policy trained with state-wise constraints is also safe in the CMDP sense; however, the converse is not assured [49]. A large body of literature on model-based CMDPs focuses on empirically reducing constraint violations [9, 50], while theoretical guarantees are often limited to simplified settings such as finite (tabular) MDPs [10, 51]. Much of this work primarily addresses safety after learning (see a survey [11]), with a few exceptions. For example, [13] considers state-wise constraints and provides safety guarantees during exploration, but only within the tabular setting. In contrast, [17] considers continuous spaces and proposes an algorithm that first uses pessimism for full exploration and then optimism for maximization of rewards in the explored region. However, all of these methods rely on *episodic* environment resets in CMDPs, making their applicability to real-world scenarios challenging.

Safety augmentation. There are also safety layer techniques that modify a standard (unconstrained) RL algorithm to enforce safety, such as safety filters [32], control barrier functions [55], and shielding [56]. Although these approaches can sometimes also ensure safety without resets, they usually cannot maintain the optimality and regret guarantees of the original RL algorithm.

Safe exploration for unknown constraints. Another line of work focuses on guaranteed exploration with unknown safety-critical constraints [12, 14, 15, 36, 57], primarily in finite (discrete) input spaces, and is typically used for safe Bayesian optimization (BO). The methodology has been applied to policy tuning [16], cost function tuning [58], and more recently scaled to continuous spaces [59] and continuous dynamical systems [18]. However, a common assumption in these approaches is to exactly choose (and reach) where to collect information. This is particularly relevant, as uncertain dynamics make it impossible to exactly reach the desired informative states, since we cannot precisely control the unknown system. Our approach addresses this problem using optimistic exploration objectives and the analysis in Proposition 2.

Model predictive control. The problem of ensuring safety when controlling an uncertain dynamical system is commonly studied in the field of optimal control using model predictive control (MPC) [60]. MPC-based methods use online numerical optimization and model-based predictions to ensure safe operation. These approaches typically do not rely on resets, but ensure persistent safety with a receding horizon implementation and a terminal set constraint, which is similar to the considered safe set $\mathbb{X}_n$ (Assumption 3) [60]. To handle model uncertainty, robust and stochastic MPC techniques predict robust or probabilistic reachable sets to ensure safe operation, see [21, 22, 42, 45, 46, 61]. In addition, to improve performance, online model updates are also incorporated in the framework of learning-based MPC or robust adaptive MPC, which typically uses dynamics model sets similar to $\mathcal{F}_n$ in Lemma 1, see [19, 20, 24, 25, 43]. In general, these approaches can ensure safe operation, but typically lack the exploration guarantees needed to achieve Objective 1 and Objective 2 (optimality results). The problem of active learning in MPC is largely based on heuristics and lacks strong guarantees of exploration; see the review papers [23, 62]. Closer to our work, [24] proposed a sequential exploration-exploitation framework that ensures safety, but the proposed active learning component provides no guarantees on uncertainty reduction, and the analysis simply assumes that the task is eventually feasible. In [25], an approach is developed that also provides exploration guarantees. However, the theoretical guarantees are only asymptotic and cannot account for probabilistic noise. Furthermore, while Proposition 2 shows that we can ensure informative measurements based on optimistic exploration objectives, this approach relies on a worst-case exploration objective, i.e., the more uncertain the system is, the less the approach can provide informative data.

B Proof for safe dynamics exploration

All theoretical results leverage the uncertainty bound from Lemma 1, which holds jointly with probability $1-\delta$, and hence all the intermediate claims are only valid with the same probability of $1-\delta$. For simplicity of exposition, we will not state the probabilistic nature of the guarantees in the intermediate lemmas and propositions.

Given the dataset $\mathcal{D}$, the posterior mean $\mu_{n,i}(z)$ and variance $\sigma^2_{n,i}(z)$ of unknown dynamics at test inputs z, z' for any component $i \in \mathbb{N}_{[1,n_x]}$ are given by:

$$\mu_{n,i}(z) = \boldsymbol{\alpha}_i^\top \boldsymbol{k}^i_{D_n}(z), \tag{13}$$

$$k^i_{D_n}(z,z') = k^i(z,z) - \boldsymbol{k}^{i\top}_{D_n}(z)(K^i_{D_n} + \sigma^2 I)^{-1}\boldsymbol{k}^i_{D_n}(z'), \tag{14}$$

$$\sigma^2(z) = k^i_{D_n}(z,z), \tag{15}$$

where $\boldsymbol{\alpha}_i = [\alpha^i_1, \dots, \alpha^i_{D_n}]^\top := (K^i_{D_n} + \sigma^2 I)^{-1} Y^i$, $\boldsymbol{k}^i_{D_n}(z) = [k^i(z_1,z), ..., k^i(z_D,z)]^\top$, where k^i is the kernel (Assumption 1), $Y^i := e_i^\top Y$ are the measurements of the i^{th} GP component ($e_i \in \mathbb{R}^{n_x}$ is a basis vector) and $K^i_{D_n}$ is the resulting kernel matrix $[k^i_{D_n}(z,z')]_{z,z' \in Z}$. For ease of notation, we use the shorthand $w_n(x,u) := \max_{i \in \mathbb{N}_{[1,n]}} w_{n,i}(x,u)$ to denote the maximal uncertainty across all state components.

Choice of tolerance $\epsilon, \epsilon_d, \epsilon_c$. As mentioned in Section 3, the tolerance $\epsilon > 0$ is a user-chosen constant that can in principle be chosen arbitrarily small, subject to a constraint related to the noise magnitude $\tilde{\eta}$. In particular, the constant $\epsilon_c > 0$ determines the threshold for informative measurements that are used to update the GP and can be chosen to an arbitrarily small positive constant. The threshold ϵ_d used for planning and termination in Algorithm 1 can then be chosen according to (16). Lastly, the tolerance $\epsilon > 0$ for the theoretical guarantees in Theorem 1 can be chosen according to (27) and (28), where $\epsilon_u \geq 0$ is arbitrarily small. In combination, these formulas provide a lower bound on the achievable guarantees $\epsilon > 0$ that depends linearly on the noise magnitude $\tilde{\eta}$.

Proof sketch. In Theorem 1, safety is ensured by always executing policies from the pessimistic safe set. We then show that only a finite number of samples, $n^\star$, can be collected before the model uncertainty falls below the ϵ_d threshold, building on the analysis in [18, 28]. While each iteration of Problem (8) identifies only a single informative state, collecting just one measurement per trajectory would be inefficient. Therefore, we extend the analysis to collect multiple measurements along a single trajectory, without requiring a GP (posterior) update at each step. A key challenge in our setting is that these measurements must be acquired by actively steering the unknown dynamical system, cf. Fig. 2. Once the model uncertainty in the pessimistic set is small, the uncertainty close to the optimistic trajectory is also small, which implies full exploration, i.e., Objective 1. The complete proof is below.

We will first prove that the optimistic and pessimistic policy sets Eqs. (6) and (7) are outer and inner approximations of the true policy set (5), respectively.

Proposition 1. *Suppose* $\boldsymbol{f}^\star \in \mathcal{F}_n$ *then* $\forall n \in \mathbb{N}, X \subseteq \mathcal{X}$ *it holds that* $\Pi^{\mathrm{p}}_n(X;\mathrm{H}) \subseteq \Pi^\star_n(X;\mathrm{H}) \subseteq \Pi^{\mathrm{o}}_n(X;\mathrm{H})$.

Proof. Consider some $\pi^{\mathrm{p}} \in \Pi^{\mathrm{p}}_n(X;\mathrm{H})$. Then by definition (6), $\exists x_0 \in X$: $x_h \in \mathcal{X}$, $u_h := \pi^{\mathrm{p}}_h(x_h) \in \mathcal{U}$, $x_\mathrm{H} \in \mathbb{X}_n, \forall \boldsymbol{f} \in \mathcal{F}_n, \forall \eta_h \in \mathcal{W}, \forall h \in \mathbb{N}_{[0,\mathrm{H}-1]}$ with $x_{h+1} = \boldsymbol{f}(x_h,u_h) + \eta_h$.

Note the difference in dynamics used in the definition of the true policy set (5) and the pessimistic policy set (6). Since the above expression holds for all $\boldsymbol{f} \in \mathcal{F}_n$ and thus also for $\boldsymbol{f}^\star \in \mathcal{F}_n$, it follows that $\pi^{\mathrm{p}} \in \Pi^\star_n(X;\mathrm{H})$. This proves $\Pi^{\mathrm{p}}_n(X;\mathrm{H}) \subseteq \Pi^\star_n(X;\mathrm{H})$.

Similarly, suppose $\pi^\star \in \Pi^\star_n(X;\mathrm{H})$. Then,

$$\exists x_0 \in X : x_h \in \mathcal{X}, u_h := \pi^\star_h(x_h) \in \mathcal{U}, x_\mathrm{H} \in \mathbb{X}_n, \forall \eta_h, \in \mathcal{W}, \forall h \in \mathbb{N}_{[0,\mathrm{H}-1]}$$

with $x_{h+1} = \boldsymbol{f}^\star(x_h,u_h) + \eta_h$. Since $\boldsymbol{f}^\star \in \mathcal{F}_n$, there exists a function $\boldsymbol{f} = \boldsymbol{f}^\star \in \mathcal{F}_n$, which satisfies the expression above and thus $\pi^\star \in \Pi^{\mathrm{o}}_n(X;\mathrm{H})$ by definition (7). □

The following proposition shows that the proposed sampling strategy always yields at least one informative sample $|\mathcal{D}_c| \geq 1$, validating the intuition from Fig. 2.

Proposition 2. *Suppose Assumption 2 holds and $\boldsymbol{f}^\star \in \mathcal{F}_n$. Let π^{p} be a feasible solution to the safe exploration Problem* (8) *for some $x(k) \in \mathcal{X}$, $n \in \mathbb{N}$ with*

$$\epsilon_d \coloneqq \epsilon_c + L_w L_{w,\mathrm{H}_c}(\epsilon_c + \tilde{\eta}), \tag{16}$$

where $L_{w,\mathrm{H}_c} = \sum_{k=0}^{\mathrm{H}_c-1}(L+L_w)^k$. Then, $\exists h \in \mathbb{N}_{[0,\mathrm{H}_c-1]}, i \in \mathbb{N}_{[1,n_x]} : w_{n,i}(x^\star_h, u^\star_h) \geq \epsilon_c$ where $x^\star_{h+1} = \boldsymbol{f}^\star(x^\star_h, u^\star_h) + \eta_h$ with $u^\star_h = \pi^{\mathrm{p}}_h(x^\star_h), \eta_h \in \mathcal{W}$, $x^\star_0 = x(k)$.

Proof. We prove this by contradiction. Suppose $\pi^{\mathrm{p}} \in \Pi^{\mathrm{p}}_n(x(k); \mathrm{H}_c)$ is a feasible solution of Problem (8) and under this policy $w_{n,i}(x^\star_h, u^\star_h) < \epsilon_c, \forall h \in \mathbb{N}_{[0,\mathrm{H}_c-1]}, i \in \mathbb{N}_{[1,n_x]}$. Let $x_{h+1} = \boldsymbol{f}(x_h, u_h)$ with $u_h = \pi^{\mathrm{p}}_h(x_h)$ and $x(k) = x_0 = x^\star_0$ be the corresponding noise-free trajectory with some $\boldsymbol{f} \in \mathcal{F}_n$. Lipschitz continuity (Assumption 2) implies

$$\begin{aligned} w_{n,i}(x_h, \pi^{\mathrm{p}}(x_h)) &\leq w_{n,i}(x^\star_h, \pi^{\mathrm{p}}(x^\star_h)) + L_w\|x_h - x^\star_h\| \\ &< L_w\|x_h - x^\star_h\| + \epsilon_c, \end{aligned} \tag{17}$$

$$\begin{aligned} \|x_{h+1} - x^\star_{h+1}\| &= \|\boldsymbol{f}(x_h, \pi^{\mathrm{p}}(x_h)) - \boldsymbol{f}^\star(x^\star_h, \pi^{\mathrm{p}}(x^\star_h)) - \eta_h\| \\ &\leq \|\eta_h\| + \|\boldsymbol{f}^\star(x_h, \pi^{\mathrm{p}}(x_h)) - \boldsymbol{f}^\star(x^\star_h, \pi^{\mathrm{p}}(x^\star_h))\| + \|\boldsymbol{f}(x_h, \pi^{\mathrm{p}}(x_h)) - \boldsymbol{f}^\star(x_h, \pi^{\mathrm{p}}(x_h))\| \\ &\overset{(4)}{\leq} \tilde{\eta} + L\|x_h - x^\star_h\| + w_{n,i}(x_h, \pi^{\mathrm{p}}(x_h)), \\ &\overset{(17)}{<} \tilde{\eta} + L\|x_h - x^\star_h\| + L_w\|x_h - x^\star_h\| + \epsilon_c, \end{aligned} \tag{18}$$

which, on using recursion, implies,

$$\|x_h - x^\star_h\| < \underbrace{\sum_{k=0}^{h-1}(L+L_w)^k}_{=:L_{w,h}}(\epsilon_c + \tilde{\eta}). \tag{19}$$

Considering the point $h' \in \mathbb{N}_{[0,\mathrm{H}_c-1]}$ with $w_{n,i}(x_{h'}, u_{h'}) \geq \epsilon_d$ from Problem (8), we have

$$\begin{aligned} w_{n,i}(x^\star_{h'}, u^\star_{h'}) &\geq w_{n,i}(x_{h'}, u_{h'}) - L_w\|x_{h'} - x^\star_{h'}\| \\ &\geq \epsilon_d - L_w L_{w,h'}(\epsilon_c + \tilde{\eta}) \\ &\geq \epsilon_d - L_w L_{w,\mathrm{H}_c}(\epsilon_c + \tilde{\eta}) \overset{(16)}{=} \epsilon_c, \end{aligned}$$

which yields a contradiction. □

To study sample complexity, we denote by d_n the number of new measurements (of all n_x components) collected to be used for the n^{th} model update. Note that Proposition 2 ensures $d_n \geq 1, \forall n \in \mathbb{N}$. Define $D_n = \sum_{j=1}^{n} d_j + D_0$ with D_0 as the prior data. If there are no prior data, $D_0 = 0$. Furthermore, we define the posterior kernel matrix $k^i_{D_{n-1}}(\cdot)$ evaluated for the d_n measurements corresponding to the i-th GP by $K^i_{d_n,d_n} \in \mathbb{R}^{d_n \times d_n}$ and the eigenvalues of this matrix by $\lambda_{r,n,i}$, $r = 1, \ldots, d_n$.

The following lemma provides a sample complexity bound, ensuring that Algorithm 1 terminates in finite time.

Lemma 2 (Sample complexity, adapted from [18])**.** *Let Assumptions 1 to 4 hold and $n^\star$ be the largest integer satisfying $\frac{n^\star}{\beta_{n^\star}\gamma_{n^\star}} \leq \frac{C_1}{\epsilon_c^2}$ with $C_1 = 8\mathrm{H}_c/\log(1 + \mathrm{H}_c\sigma^{-2})$. Consider Algorithm 1, which collects measurements whenever $w_{n,i}(z(k)) \geq \epsilon_c$. There exists an iteration $\bar{n} \leq n^\star$, such that at the end of the iteration with $x(k) \in \mathbb{X}_n$ satisfies:*

$$w_{\bar{n},i}(x_h, u_h) < \epsilon_d, \forall i \in \mathbb{N}_{[1,n_x]}, h \in \mathbb{N}_{[0,\mathrm{H}_c-1]}, \boldsymbol{f} \in \mathcal{F}_n, \pi \in \Pi^{\mathrm{p}}_{\bar{n}}(x(k), \mathrm{H}_c) \tag{20}$$

where $x_{h+1} = \boldsymbol{f}(x_h, u_h), u_h = \pi_h(x_h)$ and $x_0 = x(k)$, i.e., Algorithm 1 terminates in $\bar{n}$ iterations.

Proof. This analysis follows from [28], and we show that only a limited number of measurements can be gathered until (8) becomes infeasible. As defined earlier, confidence width

$w_{n,i}(z) = 2\sqrt{\beta_n}\sigma_{n,i}(z)$. First, note that a finite $n^\star$ exists due to the assumed sublinear growth of $\beta_n\gamma_n$, cf. Assumption 4.

A key difference from [18, 28] is that we consider n_x vector valued GPs and allow updating the model with up to H_c measurements simultaneously, provided all satisfy $w_{n-1,i}(z(k)) \geq \epsilon_c$ for the n^{th} model update. Following analysis from [15], we can bound the uncertainty at location $z_{d,n,i}$ after $n-1$ model updates as,

$$\begin{aligned}
\sum_{i=1}^{n_x}\sum_{d=1}^{d_n} w_{n-1,i}^2(z_{d,n}) &= \sum_{i=1}^{n_x}\sum_{d=1}^{d_n} 4\beta_n\sigma_{n-1,i}^2(z_{d,n}) \\
&= 4\beta_n \sum_{i=1}^{n_x} \mathrm{trace}(K_{d_n,d_n}^i) \\
&= 4\beta_n \sum_{i=1}^{n_x}\sum_{d=1}^{d_n} \lambda_{d,n,i} \\
&\overset{\text{(i)}}{\leq} 4\beta_n \sum_{i=1}^{n_x}\sum_{d=1}^{d_n} \sigma^2 C_2 \log(1+\sigma^{-2}\lambda_{d,n,i}) \\
&\leq C_1\beta_n \sum_{i=1}^{n_x}\sum_{d=1}^{d_n} \frac{1}{2}\log(1+\sigma^{-2}\lambda_{d,n,i}).
\end{aligned} \tag{21}$$

Step (i) uses the fact that $s \leq C_2\log(1+s)$ where $C_2 = \mathrm{H}_c\sigma^{-2}/\log(1+\mathrm{H}_c\sigma^{-2}) \geq 1$ $\forall s \in [0, \sigma^{-2}\mathrm{H}_c]$ with $s = \sigma^{-2}\lambda_{d,n,i}$. Note that $\lambda_{d,n,i} \leq \mathrm{trace}(K_{d_n,d_n}^i) \leq d_n \leq \mathrm{H}_c$, using the fact that $k_{D_n}(x,x) \leq k(x,x) \leq 1$, see Assumption 1. Finally, substituting $C_1 = 8\mathrm{H}_c/\log(1+\mathrm{H}_c\sigma^{-2})$ results in Eq. (21).

The sampling rule (8), i.e., collecting measurements if any one of the components has uncertainty above ϵ_c, implies

$$\sum_{d=1}^{d_n}\epsilon_c^2 \leq \sum_{i=1}^{n_x}\sum_{d=1}^{d_n} w_{n-1,i}^2(z_{d,n}) \leq C_1\beta_n \sum_{i=1}^{n_x}\sum_{d=1}^{d_n}\frac{1}{2}\log(1+\sigma^{-2}\lambda_{d,n,i})). \tag{22}$$

Analogous to [18], suppose we sample for $\bar{n}$ iterations before terminating, which implies

$$\sum_{n=1}^{\bar{n}} d_n\epsilon_c^2 \leq C_1 \sum_{n=1}^{\bar{n}}\sum_{i=1}^{n_x}\beta_{n,i}\sum_{d=1}^{d_n}\frac{1}{2}\log(1+\sigma^{-2}\lambda_{d,n,i})) \leq C_1\beta_{\bar{n}} I(Y_{\mathcal{D}_{\bar{n}}}; \boldsymbol{f}^\star_{\mathcal{D}_{\bar{n}}})) \leq C_1\beta_{\bar{n}}\gamma_{\bar{n}}, \tag{23}$$

where we used monotonicity of $\beta_{n,i}, \forall i \in \mathbb{N}_{[1,n_x]}$, denote $\beta_n = \max_i \beta_{n,i}$, the definition of the maximal information capacity γ_n and the mutual information according to Lemma 6:

$$I(Y_{\mathcal{D}_{\bar{n}}}; \boldsymbol{f}^\star_{\mathcal{D}_{\bar{n}}}) = \sum_{n=0}^{\bar{n}}\sum_{i=1}^{n_x}\sum_{d=1}^{d_n}\frac{1}{2}\log(1+\sigma^{-2}\lambda_{d,n,i})).$$

This implies

$$\frac{\bar{n}}{\beta_{\bar{n}}\gamma_{\bar{n}}} \leq \frac{1}{\beta_{\bar{n}}\gamma_{\bar{n}}}\sum_{n=1}^{\bar{n}} d_n \leq \frac{C_1}{\epsilon_c^2} \tag{24}$$

which ensures $\bar{n} \leq n^\star$ by definition of $n^\star$. Note that the condition (20) in the lemma corresponds to the termination criterion in Algorithm 1 and hence the algorithm terminates after at most $\bar{n}$ iterations. □

The following intermediate lemma uses Lipschitz continuity (Assumption 2) to bound the effect of different dynamics and policies and derive the tolerance ϵ.

Lemma 3. *Let Assumptions 1 to 4 hold. Then Algorithm 1 terminates at some state* $x(k) = x_s \in \mathbb{X}_{\bar{n}}$ *with* $\bar{n} \leq n^\star$. *Consider two policies* $\pi^b \in \Pi_{c,\bar{n}}^{\mathrm{o},\frac{\epsilon}{2}}(x_s; \mathrm{H}+\delta h)\backslash\Pi_{\bar{n}}^{\mathrm{p}}(x_s;\mathrm{H}+\delta h)$, $\pi^{\mathrm{p}} \in \Pi_{\bar{n}}^{\mathrm{p}}(x_s;\mathrm{H}+\delta h)$ *such that* $\|\pi^b(x) - \pi^{\mathrm{p}}(x)\| \leq \epsilon_u \;\forall x \in \mathcal{X}$ *with arbitrary small* $\epsilon_u > 0$ *and some* $\delta h \in \mathbb{N}_{[0,\Delta\mathrm{H}]}$. *Then it holds that* $\forall \boldsymbol{f}_1, \boldsymbol{f}_2 \in \mathcal{F}_{\bar{n}}$*:* $\|x_h^s - x_h^o\| \leq \epsilon/2$, $\|u_h^s - u_h^o\| \leq \epsilon/2, \forall h \in \mathbb{N}_{[0,\mathrm{H}+\delta h-1]}$, *with* $x_{h+1}^s = \boldsymbol{f}_1(x_h^s, u_h^s) + \eta_h$, $u_h^s := \pi^b(x_h^s)$ *and* $x_{h+1}^o = \boldsymbol{f}_2(x_h^o, u_h^o)) + \eta_h, u_h^o := \pi^b(x_h^o), x_0^o = x_0^s = x_s$, $\eta_h \in \mathcal{W}$.

Proof. From the termination criterion, we know that $w_{\bar{n},i}(x_h, u_h) < \epsilon_d, \forall h \in \mathbb{N}_{[0,\mathrm{H}_c-1]}$, $\boldsymbol{f} \in \mathcal{F}_{\bar{n}}$, $\pi \in \Pi^{\mathrm{p}}_{\bar{n}}(\mathbb{X}_{\bar{n}}; \mathrm{H}_c)$ with $x_{h+1} = \boldsymbol{f}(x_h, u_h)$, $u_h = \pi_h(x_h)$, $x_0 = x_s \in \mathbb{X}_n$. Using the invariance property of Assumption 3, we know that for all $\delta h \in \mathbb{N}_{[0,\Delta\mathrm{H}]}, \pi^{\mathrm{p}} \in \Pi^{\mathrm{p}}_{\bar{n}}(x_s; \mathrm{H} + \delta h),\ \exists \pi_f \in \Pi_{\Delta\mathrm{H}-\delta h} : [\pi^{\mathrm{p}}, \pi_f] \in \Pi^{\mathrm{p}}_{\bar{n}}(x_s; \mathrm{H}_c)$, i.e., after reaching in the terminal set with π^{p}, the appended policy π_f ensures that $\forall \boldsymbol{f} \in \mathcal{F}_{\bar{n}}$ with noise, the propagated state still remains in the safe set. Hence, $w_{n,i}(x_h, u_h) < \epsilon_d, \forall h \in \mathbb{N}_{[0,\mathrm{H}+\delta h-1]}$, $\boldsymbol{f} \in \mathcal{F}_{\bar{n}}$, $\pi \in \Pi^{\mathrm{p}}_{\bar{n}}(x_s; \mathrm{H} + \delta h)$ with $x_{h+1} = \boldsymbol{f}(x_h, u_h)$, $u_h = \pi_h(x_h)$, $x_0 = x_s \in \mathbb{X}_{\bar{n}}$ and thus $w_n(x^{\star,p}_h) < \epsilon_d\ \forall h \in \mathbb{N}_{[0,\mathrm{H}+\delta h]}$.

Define the true dynamics subject to the two different policies by $x^{\star,b}_{h+1} \coloneqq \boldsymbol{f}^\star(x^{\star,b}_h, \pi^b_h(x^{\star,b}_h)) + \eta_h$ and $x^{\star,p}_{h+1} \coloneqq \boldsymbol{f}^\star(x^{\star,p}_h, \pi^{\mathrm{p}}_h(x^{\star,p}_h))\ \forall h \in \mathbb{N}_{[0,\mathrm{H}+\delta h-1]}$, where only $x^{\star,b}_h$ is subject to noise η_h.

The remainder of this proof repeatedly uses Lipschitz continuity arguments similar to Proposition 2. The true dynamics subject to the two policies satisfies

$$\|x^{\star,b}_{h+1} - x^{\star,p}_{h+1}\| \leq L\|x^{\star,b}_h - x^{\star,p}_h\| + L_{\mathrm{f}}\epsilon_u + \tilde{\eta} \leq \cdots \leq \sum_{j=0}^{h} L^j (L_{\mathrm{f}}\epsilon_u + \tilde{\eta}), \tag{25}$$

$$\begin{aligned}
w_{\bar{n}}(x^{\star,b}_h, \pi^b_h(x^{\star,b}_h)) - & w_{\bar{n}}(x^{\star,p}_h, \pi^{\mathrm{p}}_h(x^{\star,p}_h)) \\
&\leq L'_w\|x^{\star,b}_h - x^{\star,p}_h\| + L'_w\|\pi^b_h(x^{\star,b}_h) - \pi^b_h(x^{\star,p}_h) + \pi^b_h(x^{\star,p}_h) - \pi^{\mathrm{p}}_h(x^{\star,p}_h)\| \\
&\leq L'_w(1 + L_\pi)\|x^{\star,b}_h - x^{\star,p}_h\| + L'_w\epsilon_u \\
&\overset{(25)}{\leq} \underbrace{L'_w\left(1 + L_{\mathrm{f}}(1 + L_\pi)\sum_{j=0}^{h-1} L^j\right)}_{=:K_{\epsilon,h}} \epsilon_u + \underbrace{L_w \sum_{j=0}^{h-1} L^j}_{=L_w L_h} \tilde{\eta},
\end{aligned}$$

which ensures

$$w_{\bar{n}}(x^{\star,b}_h, \pi^b_h(x^{\star,b}_h)) \leq w_{\bar{n}}(x^{\star,p}_h, \pi^p_h(x^{\star,b}_h)) + K_{\epsilon,h}\epsilon_u + L_w L_h\tilde{\eta} < \epsilon_d + K_{\epsilon,h}\epsilon_u + L_w L_h\tilde{\eta}.$$

Next, we bound the difference between the true dynamics and some dynamics $\boldsymbol{f}_1 \in \mathcal{F}_n$, both subject to the same policy π^b and same noise η_h:

$$\begin{aligned}
w_{\bar{n}}(x^s_h, \pi^b_h(x^s_h)) &\leq w_{\bar{n}}(x^{\star,b}_h, \pi^b_h(x^{\star,b}_h)) + L_w\|x^{\star,b}_h - x^s_h\| \\
&< \epsilon_d + K_{\epsilon,h}\epsilon_u + L_w L_h\tilde{\eta} + L_w\|x^{\star,b}_h - x^s_h\|, \qquad (26) \\
\|x^{\star,b}_{h+1} - x^s_{h+1}\| &\leq L\|x^{\star,b}_h - x^s_h\| + w_{\bar{n}}(x^s_h, \pi^b_h(x^s_h)) \\
&\overset{(26)}{\leq} (L + L_w)\|x^{\star,b}_h - x^s_h\| + \epsilon_d + K_{\epsilon,h}\epsilon_u + L_w L_h\tilde{\eta},
\end{aligned}$$

which using recursion until $x^{\star,b}_0 = x^s_0$ implies,

$$\|x^{\star,b}_h - x^s_h\| \leq \sum_{j=0}^{h-1} (L + L_w)^j (\epsilon_d + K_{\epsilon,h-j-1}\epsilon_u + L_w L_{h-j-1}\tilde{\eta}).$$

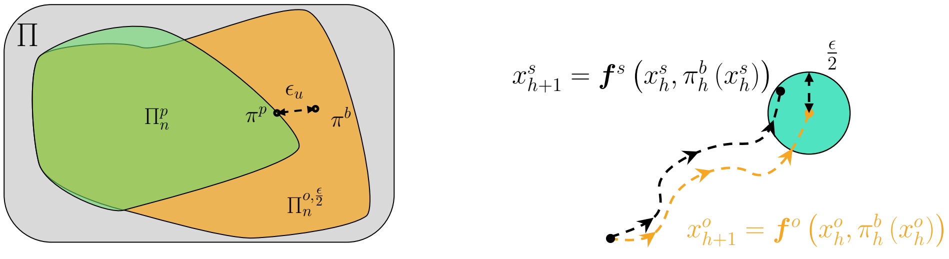
Analogously, we have for x^o with dynamics $\boldsymbol{f}_2 \in \mathcal{F}_n$:

$$\|x^{\star,b}_h - x^o_h\| \leq \sum_{j=0}^{h-1} (L + L_w)^j (\epsilon_d + K_{\epsilon,h-j-1}\epsilon_u + L_w L_{h-j-1}\tilde{\eta}),$$

which yields

$$\|x^s_h - x^o_h\| \leq 2\sum_{j=0}^{h-1} (L + L_w)^j (\epsilon_d + K_{\epsilon,h-j-1}\epsilon_u + L_w L_{h-j-1}\tilde{\eta}) \leq \epsilon/2,$$

$$\|u^s_h - u^o_h\| \leq 2L_\pi \sum_{j=0}^{h-1} (L + L_w)^j (\epsilon_d + K_{\epsilon,h-j-1}\epsilon_u + L_w L_{h-j-1}\tilde{\eta}) \leq \epsilon/2$$


Π
Π_n^p
π^p
ϵ_u
π^b
$\Pi_n^{o,\frac{\epsilon}{2}}$
$x_{h+1}^s = \boldsymbol{f}^s\left(x_h^s, \pi_h^b\left(x_h^s\right)\right)$
$\frac{\epsilon}{2}$
$x_{h+1}^o = \boldsymbol{f}^o\left(x_h^o, \pi_h^b\left(x_h^o\right)\right)$

(a) Proof by contradiction: pick $\pi^b : \|\pi^b - \pi^{\mathrm{p}}\| < \epsilon_u$ (b) Closeness of trajectories from $\boldsymbol{f}^s \in \mathcal{F}_{\bar{n}}$ under π^b

Figure 5: Illustrations of the ingredients required in the proof of Objective 1. As shown in Fig. 5a, we pick an optimistically safe policy π^b arbitrarily close to the boundary of the pessimistic policy set in a proof by contradiction. As highlighted in Fig. 5b, we show that applying the policy π^b will keep all the $\boldsymbol{f}^s \in \mathcal{F}_{\bar{n}}$ in a small ball of radius $\epsilon/2$ around the optimistic trajectory.

with

$$\epsilon > \max\{1, L_\pi\} 4 \sum_{j=0}^{\mathrm{H}_c - 1} (L + L_w)^j (\epsilon_d + L_w L_{\mathrm{H}_c - j - 1} \tilde{\eta}), \tag{27}$$

and $\epsilon_u > 0$ sufficiently small. □

The following lemma relates the termination criterion in Algorithm 1 to the uncertainty in the optimistic constraint set to prove exploration of the optimistic set $\Pi_{c,\bar{n}}^{\mathrm{o},\frac{\epsilon}{2}}(x_s; \mathrm{H} + \delta h)$.

Lemma 4. *Let Assumptions 1 to 4 hold. Then, Algorithm 1 terminates in $\bar{n}$ iteration and the current state $x(k) = x_s \in \mathbb{X}_{\bar{n}}$ satisfies $\Pi_{c,\bar{n}}^{\mathrm{o},\frac{\epsilon}{2}}(x_s; \mathrm{H} + \delta h) \subseteq \Pi_{\bar{n}}^{\mathrm{p}}(x_s; \mathrm{H} + \delta h)$.*

Proof. For contradiction, assume $\Pi_{c,\bar{n}}^{\mathrm{o},\frac{\epsilon}{2}}(x_s; \mathrm{H}+\delta h) \backslash \Pi_{\bar{n}}^{\mathrm{p}}(x_s; \mathrm{H}+\delta h) \neq \emptyset$. Due to path connectedness of $\Pi_{c,\bar{n}}^{\mathrm{o},\frac{\epsilon}{2}}(x_s; \mathrm{H} + \delta h)$ (cf. Fig. 1b), $\exists \pi^b \in \Pi_{c,\bar{n}}^{\mathrm{o},\frac{\epsilon}{2}}(x_s; \mathrm{H} + \delta h) \backslash \Pi_{\bar{n}}^{\mathrm{p}}(x_s; \mathrm{H} + \delta h)$ arbitrary close to the boundary of $\Pi_{\bar{n}}^{\mathrm{p}}(x_s; \mathrm{H}+\delta h)$ (cf. Fig. 5a). Hence,we consider a policy π^b such that $\|\pi^b(x) - \pi^{\mathrm{p}}(x)\| \leq \epsilon_u$ with $\epsilon_u > 0$ arbitrarily small.

Now, since $\pi^b \in \Pi_{c,\bar{n}}^{\mathrm{o},\frac{\epsilon}{2}}(x_s; \mathrm{H} + \delta h)$, this implies $\exists \boldsymbol{f}^o \in \mathcal{F}_{\bar{n}} : x_h^o \in \mathcal{X} \ominus \mathcal{B}_{\frac{\epsilon}{2}}, u_h \in \mathcal{U} \ominus \mathcal{B}_{\frac{\epsilon}{2}}, \forall h \in \mathbb{N}_{[0,\mathrm{H}+\delta h-1]}, \eta_h \in \mathcal{W}$, where $x_{h+1}^o = \boldsymbol{f}^o(x_h^o, u_h) + \eta_h, u_h \coloneqq \pi_h^b(x_h^o)$ and $x_{\mathrm{H}+\delta h}^o \in \mathbb{X}_{\bar{n}} \ominus \mathcal{B}_{\frac{\epsilon}{2}}$.

From Lemma 3, we know that $\forall \boldsymbol{f}^s \in \mathcal{F}_{\bar{n}}, \eta_h \in \mathcal{W}, h \in \mathbb{N}_{[0,\mathrm{H}+\delta h-1]}, x_h^s \in x_h^o \oplus \mathcal{B}_{\frac{\epsilon}{2}}$, $u_h^s \in u_h^o \oplus \mathcal{B}_{\frac{\epsilon}{2}}$ where $x_{h+1}^s = \boldsymbol{f}^s(x_h^s, u_h^s) + \eta_h$, $u_h^s = \pi^b(x_h^s)$, $x_{h+1}^o = \boldsymbol{f}^o(x_h^o, u_h^o) + \eta_h$, $u_h^o = \pi^b(x_h^o)$ $x_s = x_0^o = x_0^s$; (as shown in Fig. 5b) and both dynamics are driven by the same noise sequence $\eta_h, h \in \mathbb{N}_{[0,\mathrm{H}+\delta h-1]}$.

This implies that $\forall \boldsymbol{f}^s \in \mathcal{F}_{\bar{n}} : x_h^s \in \mathcal{X} \ominus \mathcal{B}_{\frac{\epsilon}{2}} \oplus \mathcal{B}_{\frac{\epsilon}{2}} \subseteq \mathcal{X}, \forall h \in \mathbb{N}_{[0,\mathrm{H}+\delta h]}, x_{\mathrm{H}+\delta h}^o \in \mathbb{X}_{\bar{n}} \ominus \mathcal{B}_{\frac{\epsilon}{2}} \oplus \mathcal{B}_{\frac{\epsilon}{2}} \subseteq \mathbb{X}_{\bar{n}}$ and $\pi_h^b(x_h^s) \in \mathcal{U} \ominus \mathcal{B}_{\frac{\epsilon}{2}} \oplus \mathcal{B}_{\frac{\epsilon}{2}} \subseteq \mathcal{U}$.

Hence $\pi^b \in \Pi_{\bar{n}}^{\mathrm{p}}(\mathbb{X}_{\bar{n}}; \mathrm{H} + \delta h)$. However, this is a contradiction. This implies $\Pi_{c,\bar{n}}^{\mathrm{o},\frac{\epsilon}{2}}(x_s; \mathrm{H} + \delta h) \subseteq \Pi_{\bar{n}}^{\mathrm{p}}(x_s; \mathrm{H} + \delta h)$. □

The following lemma utilizes controllability to relate the set of safe policies with a free initial condition in the safe set $\mathbb{X}_n$ to those starting at some fixed initial condition but with a larger horizon $\mathrm{H} + \delta h$.

Lemma 5. *Let Assumptions 1 to 3 hold. $\forall x' \in \mathbb{X}_n$, $\pi \in \Pi_{c,n}^{\star,\epsilon}(x'; \mathrm{H}), \exists \hat{\pi} \in \Pi_{\delta h} : [\hat{\pi}, \pi] \in \Pi_{c,n}^{\star,\frac{\epsilon}{2}}(x_s; \mathrm{H} + \delta h)\ \forall n \in \mathbb{N}$.*

Proof. Pick $\pi \in \Pi_{c,n}^{\star,\epsilon}(x'; \mathrm{H})$ with initial state $x' = x_0^\star \in \mathbb{X}_n$. Using the controllability property in Assumption 3, $\exists \hat{\pi} \in \Pi_{\delta h}$ such that $\boldsymbol{f}^\star$ can be controlled from x' to $x_0^\star \in \mathbb{X}_n$ in steps δh, without accounting for noise η. We will prove that $[\hat{\pi}, \pi] \in \Pi_{c,n}^{\star,\frac{\epsilon}{2}}(x_s; \mathrm{H} + \delta h)$, i.e., the trajectory resulting

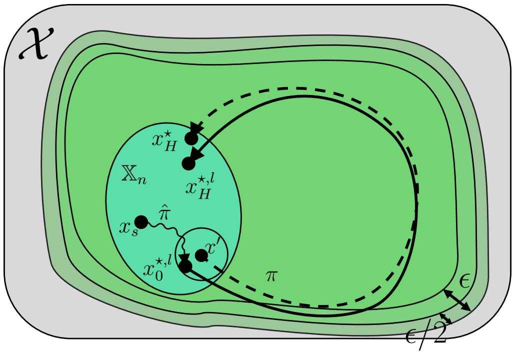


Figure 6: Illustration in state space of the trajectories used to relate the policy sets under two different horizons in Lemma 5. The policy $\pi \in \Pi^{\star,\epsilon}_{c,n}(\mathbb{X}_n; \mathrm{H})$ drives the system from a state x' to $x^\star_{\mathrm{H}} \in \mathbb{X}_n$, while ensuring that all intermediate states satisfy $x^\star_h \in \mathcal{X} \ominus \mathcal{B}_\epsilon$, $h \in \mathbb{N}_{[0,\mathrm{H}]}$ (shown with the dashed line). We show that the concatenated policy $[\hat{\pi}, \pi] \in \Pi^{\star,\frac{\epsilon}{2}}_{c,n}(x_s; \mathrm{H} + \delta h)$ control the true system $\boldsymbol{f}^\star$ starting at x_s, lands in a ball around x' (marked with circle) due to noise and then follows $x^\star_h, h \in \mathbb{N}_{[0,\mathrm{H}]}$ closely. The resulting trajectory $x^{\star,l}_h$ satisfies the constraints $x^{\star,l}_h \in \mathcal{X} \ominus \mathcal{B}_{\frac{\epsilon}{2}}$, $h \in \mathbb{N}_{[0,\mathrm{H}]}$. A similar argument applies to the input constraints, though not shown in the state space figure above.

from applying the appended policy $[\hat{\pi}, \pi]$ satisfies the constraints. Fig. 6 illustrates the two involved trajectories. Given Assumption 2, the closed-loop dynamics of $\boldsymbol{f}^\star$ with policy π is L-Lipschitz continuous. Let $x^{\star,l}_h$ and $x^\star_h$ denote the state sequences when applying the policy π and the same noise sequence η_h to dynamics $\boldsymbol{f}^\star$ with initial conditions $x^{\star,l}_0$ and $x^\star_0$, respectively. Due to noise, the deviation satisfies $\|x^\star_0 - x^{\star,l}_0\| \leq \sum_{h=0}^{\delta h - 1} L^h \tilde{\eta}$ where $x^{\star,l}_0$ is the state $\boldsymbol{f}^\star$ ends after controlling with $\hat{\pi}$. Analogous to Proposition 1, it holds that

$$\|x^\star_{h+1} - x^{\star,l}_{h+1}\| = \|\boldsymbol{f}^\star(x^\star_h, \pi_h(x^\star_h)) - \boldsymbol{f}^\star(x^{\star,l}_h, \pi_h(x^{\star,l}_h))\| \leq L\|x^\star_h - x^{\star,l}_h\| \leq L^h\|x^\star_0 - x^{\star,l}_0\|$$
$$\leq \max_{h \in \mathbb{N}_{[0,H]}} \{L^h\} \sum_{i=0}^{\delta h - 1} L^i \tilde{\eta} \leq \epsilon/2,$$

and

$$\|\pi_h(x^\star_h) - \pi_h(x^{\star,l}_h)\| \leq L_\pi \|x^\star_h - x^{\star,l}_h\| \leq \epsilon/2 \quad \forall h \in \mathbb{N}_{[0,\mathrm{H}-1]}.$$

with

$$\epsilon \geq \max\{1, L_\pi\} 2 \max\{L^{\mathrm{H}}, 1\} L_{\delta h} \tilde{\eta}. \tag{28}$$

Thus, $[\hat{\pi}, \pi] \in \Pi^{\star,\frac{\epsilon}{2}}_{c,n}(x_s; \mathrm{H} + \delta h)$. This completes the proof. □

Given, these lemmas, we are ready to prove our main result.

Theorem 1. *Let Assumptions 1 to 4 hold. Let $n^\star$ be the largest integer satisfying $\frac{n^\star}{\beta_{n^\star}\gamma_{n^\star}} \leq \frac{C_1}{\epsilon_c^2}$ with $C_1 = 8\mathrm{H}_c / \log(1 + \mathrm{H}_c \sigma^{-2})$. Then, with at least $1 - \delta$ probability, Algorithm 1 guarantees*

- *safety for all times: $x(k) \in \mathcal{X}, u(k) \in \mathcal{U}\ \forall k \in \mathbb{N}$;*
- *termination after $\bar{n} \leq n^\star$ iterations;*
- *Objective 1, ensuring maximum safe dynamics exploration.*

Proof. Safety is ensured since Algorithm 1 applies the pessimistic policy $\pi^{\mathrm{p}} \in \Pi^{\mathrm{p}}_n(x(k); \mathrm{H}_c), \forall n \geq 0$ which, by definition, ensures constraint satisfaction $\forall \boldsymbol{f} \in \mathcal{F}_n$. By Assumption 1 and Lemma 1, the unknown system satisfies $\boldsymbol{f}^\star \in \mathcal{F}_n$, thereby guaranteeing constraint satisfaction for the unknown system (1), with at least probability $1 - \delta$.

Initially at $n = 0$, Assumption 3 (control invariance) ensures that $\Pi^{\mathrm{p}}_0(x_s; \mathrm{H}_c)$ is non-empty for any $x_s \in \mathbb{X}_0$. Proposition 2 shows that the sampling rule (8) ensures collecting measurements that satisfy $w_{n,i}(z(k)) \geq \epsilon_c$. Now by Lemma 2, we know that $\exists \bar{n} \leq n^\star, x(k) \in \mathbb{X}_{\bar{n}} : \forall i \in \mathbb{N}_{[1,n_x]}, h \in \mathbb{N}_{[0,\mathrm{H}_c-1]}, \boldsymbol{f} \in \mathcal{F}_n, \pi \in \Pi^{\mathrm{p}}_{\bar{n}}(x_s, \mathrm{H}_c)$, $w_{\bar{n},i}(x_h, u_h) < \epsilon_d$ where $x_{h+1} = \boldsymbol{f}(x_h, u_h), u_h = \pi_h(x_h)$. This implies that sampling rule (8) will become infeasible in $\bar{n} \leq n^\star$ model updates, which implies that Algorithm 1 will terminate in $\bar{n}$ iterations.

Finally, Lemma 4 shows that for the final state $x(k) = x_s \in \mathbb{X}_n$ we have $\Pi^{\mathrm{o},\frac{\epsilon}{2}}_{c,\bar{n}}(x_s; \mathrm{H} + \delta h) \subseteq \Pi^{\mathrm{p}}_{\bar{n}}(x_s; \mathrm{H} + \delta h)$ which using Proposition 1 implies $\Pi^{\star,\frac{\epsilon}{2}}_{c,\bar{n}}(x_s; \mathrm{H} + \delta h) \subseteq \Pi^{\mathrm{o},\frac{\epsilon}{2}}_{c,\bar{n}}(x_s; \mathrm{H} + \delta h) \subseteq \Pi^{\mathrm{p}}_{\bar{n}}(x_s; \mathrm{H} + \delta h)$.

Thus, Lemma 5 ensures that Objective 1 holds:
$\pi^\star \in \Pi^{\star,\epsilon}_{c,\bar{n}}(\mathbb{X}_{\bar{n}}, \mathrm{H})$ implies $\exists \hat{\pi} \in \Pi_{\delta h} : [\hat{\pi}, \pi^\star] \in \Pi^{\star,\frac{\epsilon}{2}}_{c,\bar{n}}(x_s, \mathrm{H} + \delta h) \subseteq \Pi^{\mathrm{p}}_{\bar{n}}(x_s, \mathrm{H} + \delta h)$. □

The following lemma characterizes the mutual information, which is utilized for the sample complexity bound in Lemma 2.

Lemma 6 ([15, Lemma 5])**.** *Let $\mathcal{H}(Y_{\mathcal{D}_n})$ be the Shannon entropy for noisy samples $Y_{\mathcal{D}_n}$ collected at the set $\mathcal{D}_n$ [28]. Then, the mutual information $I(Y_{\mathcal{D}_{\bar{n}}}; \boldsymbol{f}^\star_{\mathcal{D}_{\bar{n}}}) := \mathcal{H}(Y_{\mathcal{D}_{\bar{n}}}) - \mathcal{H}(Y_{\mathcal{D}_{\bar{n}}} | \boldsymbol{f}^\star_{\mathcal{D}_{\bar{n}}})$ is given by*

$$I(Y_{\mathcal{D}_{\bar{n}}}; \boldsymbol{f}^\star_{\mathcal{D}_{\bar{n}}}) = \sum_{n=0}^{\bar{n}} \sum_{i=1}^{n_x} \sum_{d=1}^{d_n} \frac{1}{2} \log(1 + \sigma^{-2} \lambda_{d,n,i}))$$

Proof. The data set $\mathcal{D}_n = \{d_0, \dots, d_n\}$

$$\begin{aligned}
\mathcal{H}(Y_{\mathcal{D}_n}) &= \mathcal{H}(Y_{d_n} | Y_{\mathcal{D}_{n-1}}) + \mathcal{H}(Y_{\mathcal{D}_{n-1}}) && (29) \\
&= \sum_{i=1}^{n_x} \left(\frac{1}{2} \log(\det(2\pi e(\sigma^2 I + K^i_{d_n,d_n}))) \right) + \mathcal{H}(Y_{d_{n-1}} | Y_{\mathcal{D}_{n-1}}) + \dots && (30) \\
&= \sum_{i=1}^{n_x} \left(\frac{d_n}{2} \log(2\pi e \sigma^2) + \frac{1}{2} \log(\det(I + \sigma^{-2} K^i_{d_n,d_n}))) \right) + \mathcal{H}(Y_{d_{n-1}} | Y_{\mathcal{D}_{n-1}}) + \dots && (31) \\
&= \sum_{n=0}^{\bar{n}} \sum_{i=1}^{n_x} \frac{d_n}{2} \log(2\pi e \sigma^2) + \frac{1}{2} \log(\det(I + \sigma^{-2} K^i_{d_n,d_n}))) && (32)
\end{aligned}$$

For (30), we used that each component i is independently sampled as $Y_{d_n} \sim \mathcal{N}(Z_{d_n}, \sigma^2 I + K^i_{d_n,d_n})$ is jointly a multivariate Gaussian. Eq. (31) follows by

$$\begin{aligned}
\log\left(\det\left(2\pi e \left(\sigma^2 I + K^i_{d_n,d_n}\right)\right)\right) &= \log\left(\left(2\pi e \sigma^2\right)^{d_n} \det\left(\sigma^2 I + K^i_{d_n,d_n}\right)\right) \\
&= d_n \log\left(2\pi e \sigma^2\right) + \log\left(\det\left(\sigma^2 I + K^i_{d_n,d_n}\right)\right).
\end{aligned}$$

Finally, Eq. (32) follows by repeating the above 2 steps recursively until $n = 0$. $\mathcal{H}(Y_{\mathcal{D}_n} | \boldsymbol{f}^\star_{\mathcal{D}_n}) = \sum_{n=0}^{\bar{n}} \sum_{i=1}^{n_x} \frac{d_n}{2} \log(2\pi e \sigma^2)$ is the entropy due to noise. On substituting this together with Eq. (32) in the mutual information definition, we obtain

$$\begin{aligned}
I(Y_{\mathcal{D}_{\bar{n}}}; \boldsymbol{f}^\star_{\mathcal{D}_{\bar{n}}}) &= \frac{1}{2} \sum_{n=0}^{\bar{n}} \sum_{i=1}^{n_x} \log(\det(I + \sigma^{-2} K^i_{d_n,d_n})) \\
&\overset{\text{(i)}}{=} \frac{1}{2} \sum_{n=0}^{\bar{n}} \sum_{i=1}^{n_x} \log(\prod_{d=1}^{d_n} (1 + \sigma^{-2} \lambda_{d,n,i})) \\
&= \frac{1}{2} \sum_{n=0}^{\bar{n}} \sum_{i=1}^{n_x} \sum_{d=1}^{d_n} \log(1 + \sigma^{-2} \lambda_{d,n,i}). && (33)
\end{aligned}$$

Step (i) follows by orthonormal eigen decomposition of the kernel matrix, where $\lambda_{d,n,i}$ are the eigenvalues of $K^i_{d_n,d_n}$. □

C Reward maximization with intrinsic exploration

This section provides a more detailed explanation of the proposed reward maximization algorithm SAGEDYNX in Appendix C.1, and then we provide the proof of the theoretical guarantees in Appendix C.2.

Choice of tolerance $\epsilon, \epsilon', \epsilon_d, \epsilon_c$. As in the maximum dynamics exploration setting (Section 3), in Theorem 2, tolerance $\epsilon > 0$ can be chosen arbitrarily small, constrained only by the noise magnitude $\tilde{\eta}$. The constants ϵ_c and ϵ_d serve analogous roles — $\epsilon_c > 0$ sets the threshold for informative measurement, while ϵ_d, used in Algorithm 2 for planning is selected as in (16). The tolerance ϵ' that tightens the constraints for the pessimistic policy set can be chosen satisfying (38), (44), and (52). Lastly, the tolerance $\epsilon > 0$ for certifying Objective 2 can be chosen according to (54), and the Objective 2 holds with constant K defined as per (39). Together, these formulas provide a lower bound on the achievable performance with $\epsilon > 0$ that scales linearly with the noise magnitude $\tilde{\eta}$. In the noise-free case, $\tilde{\eta} = 0$, the tolerance $\epsilon > 0$ can be chosen arbitrarily small.

C.1 Detailed explanation on SAGEDYNX algorithm

SAGEDYNX revolves mainly around Line 8, where at each time step k, the agent solves Problem (12) using it's current state $x(k)$ and a task-driven objective J^{any}. The agent executes the resulting policy π_e^{p}, collects new measurements having sufficient information, and then updates the model. It instantaneously replans (resolves Problem (12)) with the updated model at the current state $x(k)$. As shown in Fig. 3a, a key component is that the agent always ensures the existence of a safe return path without needing to execute it explicitly. Over time, with data, the agent learns and gets closer to the optimal behavior with iterations. The agent continues to explore until the dynamics is sufficiently well learned to achieve the objective of the task. Finally, when the agent is guaranteed to achieve close-to-optimal behavior, it returns and executes the safe close-to-optimal policy.

The approach does an intrinsic exploration while maximizing a task-driven objective J^{any}. This objective can be heuristically designed, as discussed in Remark 4. It can naturally blend exploration and task-oriented behavior, for example, minimizing distance to the optimistic trajectory.

To determine when the dynamics are sufficiently learned (guaranteed to exhibit close to optimal behavior), we need a termination criterion. There are particularly two challenges for termination criteria:
i) should *not* require a uniform reduction in uncertainty (less than ϵ_d) throughout
ii) should guarantee bounded regret (with returned policy) despite not uniformly knowing the dynamics along the executed trajectory.

To overcome this, we define a termination criterion that involves solving the pessimistic problem (10) (right) with the following objective,

$$J^{\mathrm{p}}(x, \boldsymbol{\mu}_n; \pi) \coloneqq J(x, \boldsymbol{\mu}_n; \pi) - L_r \sum_{h=0}^{\mathrm{H}-1} \sum_{i=0}^{h} L^i w_n(x_i, \pi_i(x_i)), \tag{34}$$

representing a pessimistic estimate (lower bound) of rewards (c.f. Lemma 9), ensuring that on termination, even for the worst realization (in terms of objective) of the dynamics will achieve close to $J^{\mathrm{p}}(x, \boldsymbol{\mu}_n; \pi)$ of the true trajectory. With this, we define the termination criteria as follows,

$$J^{\mathrm{p}}(x^{\mathrm{p}}, \boldsymbol{\mu}_n; \pi^{\mathrm{p}}) \geq J(x^{\mathrm{o}}, \boldsymbol{f}^{o}; \pi^{\mathrm{o}}) - 3L_r \sum_{h=0}^{\mathrm{H}-1} L_h \epsilon_d, \tag{35}$$

where x^{p},π^{p} are the optimal solution of the pessimistic problem (10) (right) and x^{o}, $\boldsymbol{f}^{o}$, π^{o} are the optimal solution of the optimistic problem (10) (left). Notably, while the small uncertainty (smaller than tolerance ϵ_d) in the true trajectory after executing the returned policy is a sufficient stopping condition, the termination criterion (35) does not necessarily require this.

One may also employ alternative termination criteria, which may improve computational efficiency by involving fewer optimization problems to be solved. For example, if the model uncertainty along the optimistic trajectory is uniformly below ϵ_d, then executing the corresponding optimistic policy is sufficient to ensure that the true system performs close to the optimistic estimate—and therefore close to the optimal behaviour. This termination criterion avoids solving the pessimistic problem

(10) (right), but may require more iterations to satisfy the uncertainty condition compared to the termination criterion in (35).

While the termination criterion in (35) requires solving both the optimistic and pessimistic problems, these problems do not directly influence the exploration process and are only used for checking termination. As a result, it need not be evaluated at every iteration n; instead, it can be checked periodically or heuristically, depending on the task and available computational budget.

Remark 4 (Task-oriented exploration with $J^{\mathrm{any}}(x_s, n; \pi)$)**.** *Based on the theoretical analysis, it suffices if the uncertainty in the dynamics is small around the optimistically planned trajectory. Hence, a natural objective J^{any} to consider is aiming for informative measurements $w_n(x_h, u_h) \gg \epsilon$ that are also close to the optimistically optimal trajectory. J^{any} can be the distance to the optimistic path, the first state on the optimistic path, or cumulative rewards and uncertainty along the trajectory.*

C.2 Proof of Theorem 2

Proof sketch. Analogous to Theorem 1, safety is guaranteed by only executing pessimistically safe policies. However, it is challenging to show that once the termination criteria is satisfied, we can find a policy that satisfies Objective 2. This is due to the fact that the state reached in the safe set will be different from the state optimized in Problem (10) used to check the termination criteria. To overcome this, we use the fact that the dynamics is known up to ϵ_c tolerance in the safe set, and problem (10) in the termination criterion uses a threshold $\epsilon' \geq 0$ for the distance to the constraints. Finally, the algorithm terminates in finite time, at the latest once the uncertainty is reduced below ϵ_d everywhere, with the worst-case sample complexity shown in Theorem 1. The detailed proof is below.

The following lemma bounds the deviation between the optimistic and the true dynamics.

Lemma 7. *Let Assumption 2 hold and suppose that $\boldsymbol{f}^\star \in \mathcal{F}_n$. Consider any policy $\pi \in \Pi_{\mathrm{H}}$, dynamics $\boldsymbol{f} \in \mathcal{F}_n$, initial condition $x_0^\star = x_0^{\mathrm{o}} \in \mathcal{X}$, and the two trajectories $x_{h+1}^\star = \boldsymbol{f}^\star(x_h^\star, \pi_h(x_h^\star))$, $x_{h+1}^{\mathrm{o}} = \boldsymbol{f}(x_h^{\mathrm{o}}, \pi_h(x_h^{\mathrm{o}}))$. Then, for any $h \in \mathbb{N}$ it holds that,*

$$\|x_{h+1}^\star - x_{h+1}^{\mathrm{o}}\| \leq \sum_{i=0}^{h} L^i w_{n-1}(x_i, \pi_i(x_i)) \tag{36}$$

Proof. Consider,

$$\begin{aligned}
\|x_{h+1}^\star - x_{h+1}\| &\leq \|\boldsymbol{f}^\star(x_h^\star, \pi_h(x_h^\star)) - \boldsymbol{f}(x_h, \pi_h(x_h))\| \\
&\leq \|\boldsymbol{f}^\star(x_h^\star, \pi_h(x_h^\star)) - \boldsymbol{f}^\star(x_h, \pi_h(x_h))\| + \|\boldsymbol{f}^\star(x_h, \pi_h(x_h)) - \boldsymbol{f}(x_h, \pi_h(x_h))\| \\
&\overset{\text{(i)}}{\leq} L\|x_h^\star - x_h\| + w_{n-1}(x_h, \pi_h(x_h)) \\
&\vdots \\
&\leq \sum_{i=0}^{h} L^i w_{n-1}(x_i, \pi_i(x_i))
\end{aligned} \tag{37}$$

Step (i) follows since $\|\boldsymbol{f}^\star(x_h, \pi_h(x_h)) - \boldsymbol{f}(x_h, \pi_h(x_h))\| \leq w_n(x_h, \pi_h(x_h))$ and Lipschtiz continuity of the closed-loop system $\boldsymbol{f}^\star$. The final inequality follows using recursion. □

The following lemma demonstrates that the policy $[\hat{\pi}, \pi^{\mathrm{p}}]$ returned by Algorithm 2 ensures the safety of the unknown system.

Lemma 8 (Safety with returned policy)**.** *Let Assumptions 1 to 5 hold. Let $\hat{\pi} \in \Pi_{\delta h}$ control the mean dynamics from $x(k)$ to x^{p} and $\pi^{\mathrm{p}} \in \Pi_n^{\mathrm{p},\epsilon'}(x^{\mathrm{p}}; \mathrm{H})$ obtained by solving* (10) *(right). Then the returned policy $\pi^{\mathrm{r}} := [\hat{\pi}, \pi^{\mathrm{p}}] \in \Pi_n^\star(x(k); \mathrm{H} + \delta h)$, that is, the resulting closed-loop system from the returned policy satisfies constraints and ends in the safe set.*

Proof. Given Assumption 2, the closed-loop dynamics of $\boldsymbol{f}^\star$ with policy π is L-Lipschitz continuous. Similarly to Lemma 5, let $x_h^{\star,l}$ and $x_h^\star$ denote the state sequences when applying the policy π^{p}, and the same noise sequence η_h to the dynamics $\boldsymbol{f}^\star$ with initial conditions $x_0^{\star,l}$ and $x_0^\star$, respectively. Due to controllability within the safe set (Assumption 3), unknown dynamics up to ϵ_c tolerance in the safe set (Assumption 5) and noise, the deviation satisfies $\|x_0^\star - x_0^{\star,l}\| \le \sum_{h=0}^{\delta h-1} L^h(\epsilon_c + \tilde{\eta})$ where $x_0^{\star,l}$ is the state at which $\boldsymbol{f}^\star$ ends after controlling with $\hat{\pi}$ (policy $\hat{\pi}$ is optimized with mean dynamics $\boldsymbol{\mu}_n$).

Analogous to Proposition 1, it holds that

$$\begin{aligned}\|x_{h+1}^\star - x_{h+1}^{\star,l}\| = \|\boldsymbol{f}^\star(x_h^\star, \pi_h(x_h^\star)) - \boldsymbol{f}^\star(x_h^{\star,l}, \pi_h(x_h^{\star,l}))\| \le L\|x_h^\star - x_h^{\star,l}\| \le L^h\|x_0^\star - x_0^{\star,l}\| \\ \le \max_{h\in\mathbb{N}_{[0,H]}}\{L^h\}\sum_{i=0}^{\delta h-1} L^i(\epsilon_c + \tilde{\eta}) \le \epsilon'\end{aligned}$$

and

$$\|\pi_h(x_h^\star) - \pi_h(x_h^{\star,l})\| \le L_\pi\|x_h^\star - x_h^{\star,l}\| \le \epsilon', \quad \forall h \in \mathbb{N}_{[0,\mathrm{H}-1]}.$$

with

$$\epsilon' \ge \max\{1, L_\pi\}\max\{L^{\mathrm{H}}, 1\}L_{\Delta\mathrm{H}}(\epsilon_c + \tilde{\eta}). \tag{38}$$

Since tightening ϵ' is used in (10) (right) while optimizing for π^{p} implies $\pi^{\mathrm{p}} \in \Pi_n^{\mathrm{p},\epsilon'}(x(k);\mathrm{H}) \subseteq \Pi_n^{\star,\epsilon'}(x(k);\mathrm{H})$, the appended policy satisfies $[\hat{\pi}, \pi^{\mathrm{p}}] \in \Pi_n^\star(x(k);\mathrm{H}+\delta h)$.

Furthermore, since $[\hat{\pi}, \pi^{\mathrm{p}}] \in \Pi_n^\star(x(k);\mathrm{H}+\delta h)$, this implies that the resulting closed-loop system $\boldsymbol{f}^\star \in \mathcal{F}_n$ with $[\hat{\pi}, \pi^{\mathrm{p}}]$ satisfies constraints (2). □

Lemma 9 (Optimality with return policy). *Let Assumptions 1 to 5 hold. Suppose the termination criterion* (35) *is satisfied. Then the resulting closed-loop system from the return policy* $[\hat{\pi}, \pi^{\mathrm{p}}]$ *guarantees Objective 2 with constant*

$$K\epsilon = 3L_r\sum_{h=0}^{\mathrm{H}-1} L_h\epsilon_d + L_r L_{\mathrm{H}} L_{\delta h}(\epsilon_c + \tilde{\eta}). \tag{39}$$

Proof. We will derive a lower bound for $J(x(k), \boldsymbol{f}^\star; \pi^{\mathrm{r}})$ which is given by,

$$J(x(k), \boldsymbol{f}^\star; [\hat{\pi}, \pi^{\mathrm{p}}]) = J(x(k), \boldsymbol{f}^\star; \hat{\pi}) + \mathbb{E}_{x_0^\star\sim\hat{\pi}} J(x_0^\star, \boldsymbol{f}^\star; \pi^{\mathrm{p}}) \overset{\text{(i)}}{\ge} \mathbb{E}_{x_0^\star\sim\hat{\pi}} J(x_0^\star, \boldsymbol{f}^\star; \pi^{\mathrm{p}}), \tag{40}$$

where $x_0^\star$ is the state (random variable) of the true dynamics controlling from $x(k)$ with the policy $\hat{\pi}$ under the noise distribution which is denoted by $\mathbb{E}_{x_0^\star\sim\hat{\pi}}$. Step (i) follows from $r(\cdot,\cdot) \ge 0$.

Next, we will bound the difference in cumulative rewards between mean dynamics $\boldsymbol{\mu}_n$ and $\boldsymbol{f}^\star$, using the policy π^{p}, when starting, respectively, from x^{p} and $x_0^\star$ and realizing the same noise sequence, resulting in trajectories x_h and $x_h^\star$ respectively.

$$\begin{aligned}&J(x^{\mathrm{p}}, \boldsymbol{\mu}_n; \pi^{\mathrm{p}}) - \mathbb{E}_{x_0^\star\sim\hat{\pi}} J(x_0^\star, \boldsymbol{f}^\star; \pi^{\mathrm{p}}) \\ &\qquad = \mathbb{E}_{x_0^\star\sim\hat{\pi}}\mathbb{E}_\eta\left(\sum_{h=0}^{\mathrm{H}-1} r(x_h, \pi_h^{\mathrm{p}}(x_h)) - r(x_h^\star, \pi_h^{\mathrm{p}}(x_h^\star))\right) \\ &\qquad \le \mathbb{E}_{x_0^\star\sim\hat{\pi}}\mathbb{E}_\eta L_r\sum_{h=0}^{\mathrm{H}-1}\|x_h - x_h^\star\| \\ &\qquad \le L_r\sum_{h=0}^{\mathrm{H}-1}\left(L^h L_{\delta h}(\epsilon_c + \tilde{\eta}) + \mathbb{E}_\eta\sum_{i=0}^{h-1} L^i w_{n-1}(x_i, \pi_i^{\mathrm{p}}(x_i))\right) \\ &\qquad = L_r L_{\mathrm{H}} L_{\delta h}(\epsilon_c + \tilde{\eta}) + \mathbb{E}_\eta L_r\sum_{h=0}^{\mathrm{H}-1}\sum_{i=0}^{h-1} L^i w_{n-1}(x_i, \pi_i^{\mathrm{p}}(x_i))\end{aligned} \tag{41}$$

In the above, the first inequality follows from the Lipschitz continuity of rewards and the policies and for the second inequality, analogous to Lemma 7, we used that

$$\begin{aligned}\|x_h - x_h^\star\| &\le L\|x_{h-1}^\star - x_{h-1}\| + w_{n-1}(x_{h-1}, \pi^{\mathrm{p}}(x_{h-1})) \\ &\le L\|x_{h-2}^\star - x_{h-2}\| + Lw_{n-1}(x_{h-2}, \pi^{\mathrm{p}}(x_{h-2})) + w_{n-1}(x_{h-1}, \pi^{\mathrm{p}}(x_{h-1}))\end{aligned}$$

$$\leq L^h L_{\delta h}(\epsilon_c + \tilde{\eta}) + \sum_{i=0}^{h-1} L^i w_{n-1}(x_i, \pi^{\mathrm{p}}(x_i))$$

where the last inequality uses that $\|x_0 - x_0^\star\| \leq L_{\delta h}(\epsilon_c + \tilde{\eta})$ by Assumption 5 of knowing dynamics upto ϵ_c tolerance in the safe set. Since the upper bound is constant (does not depend on $x_0^\star$), $\mathbb{E}_{x_0^\star \sim \hat{\pi}}$ disappears.

Finally, on substituting Eq. (41) in Eq. (40), we get,

$$\begin{aligned}
J(x(k), \boldsymbol{f}^\star; [\hat{\pi}, \pi^{\mathrm{p}}]) &\geq J(x^{\mathrm{p}}, \boldsymbol{\mu}_n; \pi^{\mathrm{p}}) - \mathbb{E}_\eta L_r \sum_{h=0}^{\mathrm{H}-1} \sum_{i=0}^{h-1} L^i w_{n-1}(x_i, \pi^{\mathrm{p}}(x_i)) - L_r L_{\mathrm{H}} L_{\delta h}(\epsilon_c + \tilde{\eta}) \\
&\overset{\text{(i)}}{\geq} J(x^{\mathrm{o}}, \boldsymbol{f}^{o}; \pi^{\mathrm{o}}) - 3L_r \sum_{h=0}^{\mathrm{H}-1} L_h \epsilon_d - L_r L_{\mathrm{H}} L_{\delta h}(\epsilon_c + \tilde{\eta}) \\
&\overset{\text{(ii)}}{\geq} J(x^\star, \boldsymbol{f}^\star; \pi^\star) - 3L_r \sum_{h=0}^{\mathrm{H}-1} L_h \epsilon_d - L_r L_{\mathrm{H}} L_{\delta h}(\epsilon_c + \tilde{\eta})
\end{aligned}$$

Step (i) follows from the termination criterion (35). Step (ii) follows since the optimistic policy set is always a subset of the true policy set, see Proposition 1. The last inequality implies satisfying Objective 2 with K as per (39). □

Lemma 10. *Let Assumptions 1 to 4 hold. Let $\bar{n} \leq n^\star$ be as defined in Theorem 1. Consider any $x_0^\star \in \mathbb{X}_{\bar{n}}$, $\pi^{\mathrm{p}} \in \Pi^{\mathrm{p},\epsilon'}_{\bar{n}}(x_0^\star, \mathrm{H})$ and the resulting trajectory $x^\star_{h+1} = \boldsymbol{f}^\star(x^\star, \pi^{\mathrm{p}}(x^\star))$, $\forall h \in \mathbb{N}_{[0,\mathrm{H}-1]}$. Then the following holds:*

$$w_{\bar{n}}(x, \pi^{\mathrm{p}}_h(x)) \leq L_w \|x - x^\star_h\| + \epsilon_d, h \in \mathbb{N}_{[0,\mathrm{H}]}, x \in \mathcal{X}. \tag{42}$$

Proof. Given the sampling criteria in Algorithm 2 and using Lemma 2, we obtain $\exists \bar{n} \leq n, x_s \in \mathbb{X}_{\bar{n}}$:

$$w_{\bar{n}}(x_h, \pi^{\mathrm{p}}(x_h)) < \epsilon_d, \forall h \in \mathbb{N}_{[0,\mathrm{H}_c-1]}, \boldsymbol{f} \in \mathcal{F}_n, \pi^{\mathrm{p}} \in \Pi^{\mathrm{p}}_{\bar{n}}(x_s, \mathrm{H}_c), \tag{43}$$

where $x_{h+1} = \boldsymbol{f}(x_h, \pi^{\mathrm{p}}_h(x_h))$ with $x_0 = x_s$. Suppose (cf. Eq. (27))

$$\epsilon' > \max\{1, L_\pi\} 4 \sum_{j=0}^{\mathrm{H}_c - 1} (L_w + L)^j (\epsilon_d + L_w L_{\mathrm{H}-j-1} \tilde{\eta}) \tag{44}$$

which ensures $\Pi^{\star,\frac{\epsilon'}{2}}_{c,\bar{n}}(x_s; \mathrm{H}_c) \subseteq \Pi^{\mathrm{p}}_{\bar{n}}(x_s; \mathrm{H}_c)$ by using Lemma 4. Along with (43), this implies

$$w_{\bar{n}}(x^{\star'}_h, \pi^\star_h(x^{\star'}_h)) < \epsilon_d, \forall h \in \mathbb{N}_{[0,\mathrm{H}_c-1]}, \pi^\star \in \Pi^{\star,\frac{\epsilon'}{2}}_{c,\bar{n}}(x_s, \mathrm{H}_c), \tag{45}$$

with $x^{\star'}_{h+1} = \boldsymbol{f}(x^{\star'}_h, \pi^\star_h(x^{\star'}_h))$ with $x^{\star'}_0 = x_s$. Using the invariance property of Assumption 3, for all $\delta h \in \mathbb{N}_{[0,\Delta\mathrm{H}]}, \pi^\star \in \Pi^{\star,\frac{\epsilon'}{2}}_{c,\bar{n}}(x_s, \mathrm{H} + \delta h)$, $\exists \pi_f \in \Pi_{\Delta\mathrm{H}-\delta h}$: $[\pi^\star, \pi_f] \in \Pi^{\star,\frac{\epsilon'}{2}}_{c,\bar{n}}(x_s, \mathrm{H}_c)$ which from (45) implies,

$$w_{\bar{n}}(x^{\star'}_h, \pi^\star_h(x^{\star'}_h)) < \epsilon_d, \forall h \in \mathbb{N}_{[0,\mathrm{H}_c-1]}, \pi^\star \in \Pi^{\star,\frac{\epsilon'}{2}}_{c,\bar{n}}(x_s, \mathrm{H} + \delta h), \tag{46}$$

Since ϵ' satisfies (38), it also holds that $\epsilon' \geq \max\{1, L_\pi\} 2 \max\{L^{\mathrm{H}}, 1\} L_{\Delta\mathrm{H}} \tilde{\eta}$. Then by using Lemma 5 we get,

$$w_{\bar{n}}(x^\star_h, \pi^\star_h(x^\star_h)) < \epsilon_d, \forall h \in \mathbb{N}_{[0,\mathrm{H}-1]}, x_0^\star \in \mathbb{X}_{\bar{n}}, \pi^\star \in \Pi^{\star,\epsilon'}_{c,\bar{n}}(x_0^\star, \mathrm{H}), \tag{47}$$

with $x^\star_{h+1} = \boldsymbol{f}(x^\star_h, \pi^\star_h(x^\star_h))$. In the above, optimistic trajectory $x^\star_h$ satisfies $w_{\bar{n}}(x^\star_h, \pi^\star_h(x^\star_h)) < \epsilon_d$, since $x^{\star'}_h$ in (45) is a noiseless trajectory and thus with $\hat{\pi}$ we can control it from x_s to any location $x_0^\star \in \mathbb{X}_n$ exactly.

Using Proposition 1, we know that $\Pi^{\mathrm{p},\epsilon'}_{\bar{n}}(\mathbb{X}_{\bar{n}}, \mathrm{H}) \subseteq \Pi^{\star,\epsilon'}_{c,\bar{n}}(\mathbb{X}_{\bar{n}}, \mathrm{H})$ and thus the noise-free trajectory from true dynamics under pessimistic policy π^{p} satisfies,

$$w_{\bar{n}}(x^\star_h, \pi^{\mathrm{p}}_h(x^\star_h)) < \epsilon_d, \forall h \in \mathbb{N}_{[0,\mathrm{H}-1]}, x_0^\star \in \mathbb{X}_{\bar{n}}, \pi^{\mathrm{p}} \in \Pi^{\mathrm{p},\epsilon'}_{\bar{n}}(x_0^\star, \mathrm{H}). \tag{48}$$

Finally, using the Lipschitz continuity of w and π with any point $x \in \mathcal{X}$ yields (42), which concludes the proof. □

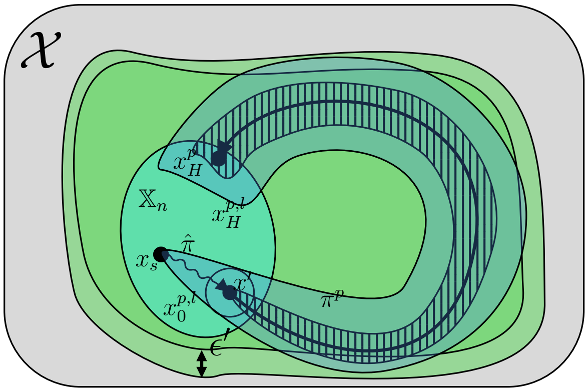


Figure 7: Illustration in state space of the trajectories used to relate the pessimistic policy sets under two different horizons in Lemma 5. The policy $\pi^{\mathrm{p}} \in \Pi_{\bar{n}}^{\mathrm{p},\epsilon'}(\mathbb{X}_n; \mathrm{H})$ drives all dynamics $\boldsymbol{f}$ from a state x' to $x_{\mathrm{H}}^{\mathrm{p}} \in \mathbb{X}_n$ (marked with shaded lines), while ensuring that all intermediate states $x_h^{\mathrm{p}} \in \mathcal{X} \ominus \mathcal{B}_{\epsilon'}$, for all $h \in \mathbb{N}_{[0,\mathrm{H}]}$. We show that the concatenated policy $[\hat{\pi}, \pi^{\mathrm{p}}] \in \Pi_{\bar{n}}^{\mathrm{p}}(x_s; \mathrm{H} + \delta h)$ control any dynamics $\boldsymbol{f} \in \mathcal{F}_{\bar{n}}$ starting at x_s to x', where all the dynamics land in a ball around x' (marked with a circle) due to unknown dynamics and noise, and then follows π^{p}. The resulting trajectories $x_h^{\mathrm{p},l}$ for all dynamics (region indicated by semi-transparent blue color) satisfy the constraints $x_h^{\mathrm{p},l} \in \mathcal{X}$ for all horizons. A similar argument applies to the input constraints, though not shown in the state space figure above.

In the following, we show that the sampling rule (8) ensures exploration of the pessimistic policy set $\Pi_n^{\mathrm{p},\epsilon'}(\mathbb{X}_n; \mathrm{H})$.

Lemma 11. *Let Assumptions 1 to 4 hold.* $\exists \bar{n} \leq n^\star : \forall x' \in \mathbb{X}_n,\ \pi^{\mathrm{p}} \in \Pi_{\bar{n}}^{\mathrm{p},\epsilon'}(x'; \mathrm{H}), \exists \hat{\pi} \in \Pi_{\delta h} :$ $[\hat{\pi}, \pi^{\mathrm{p}}] \in \Pi_{\bar{n}}^{\mathrm{p}}(x_s; \mathrm{H} + \delta h)$.

Proof. Given x_s as an initial state, any target state $x' \in \mathbb{X}_n$ and any dynamics $\boldsymbol{f} \in \mathcal{F}_n$ (optimistic), consider applying policy $\hat{\pi}$ (Assumption 3) that control this dynamics $\boldsymbol{f}$ from $x_s \to x' =: x_0^{\mathrm{p}}$. However, due to noise and unknown dynamics in the safe set, the worst deviation satisfies $\|x_0^{\mathrm{p}} - x_0^{\mathrm{p},l}\| \leq \sum_{h=0}^{\delta h - 1} L^h(\epsilon_c + \tilde{\eta})$ where $x_0^{\mathrm{p},l}$ results from any (worst-case) dynamics $\boldsymbol{f} \in \mathcal{F}_n$ under the policy $\hat{\pi}$ (cf. Lemma 8). After that, let $x_h^{\mathrm{p},l}$ and x_h^{p} denote the state sequences generated by applying the policy π^{p} and some noise sequences η_h to the dynamics $\boldsymbol{f}$ with initial conditions $x_0^{\mathrm{p},l}$ and x_0^{p}, respectively. See Fig. 7 for an illustration.

We will bound $\|x_{h+1}^{\mathrm{p},l} - x_{h+1}^{\mathrm{p}}\|, \forall h \in \mathbb{N}_{[0,\mathrm{H}-1]}$ to determine ϵ' such that the policy $[\hat{\pi}, \pi^{\mathrm{p}}]$ keeps all dynamics safe, i.e. $[\hat{\pi}, \pi^{\mathrm{p}}] \in \Pi_{\bar{n}}^{\mathrm{p}}(x_s; \mathrm{H} + \delta h)$. For this, define noiseless true trajectory sequence $x_{h+1}^\star = \boldsymbol{f}^\star(x_h^\star, \pi_h^{\mathrm{p}}(x_h^\star))$ with initial conditions $x_0^\star = x_0^{\mathrm{p}}$. Using triangle inequality, we get,

$$\|x_{h+1}^{\mathrm{p},l} - x_{h+1}^{\mathrm{p}}\| \leq \|x_{h+1}^{\mathrm{p},l} - x_{h+1}^\star\| + \|x_{h+1}^\star - x_{h+1}^{\mathrm{p}}\|. \tag{49}$$

Analogous to Proposition 2, it holds that

$$\begin{aligned}\|x_{h+1}^\star - x_{h+1}^{\mathrm{p}}\| &= \|\boldsymbol{f}^\star(x_h^\star, \pi_h^{\mathrm{p}}(x_h^\star)) - \boldsymbol{f}^\star(x_h^{\mathrm{p}}, \pi_h^{\mathrm{p}}(x_h^{\mathrm{p}}))\| + \|\boldsymbol{f}^\star(x_h^{\mathrm{p}}, \pi_h^{\mathrm{p}}(x_h^{\mathrm{p}})) - \boldsymbol{f}(x_h^{\mathrm{p}}, \pi_h^{\mathrm{p}}(x_h^{\mathrm{p}})) - \tilde{\eta}\| \\ &\overset{(42)}{\leq} L\|x_h^\star - x_h^{\mathrm{p}}\| + L_w\|x_h^{\mathrm{p}} - x_h^\star\| + \epsilon_d + \tilde{\eta} \\ &\leq (L + L_w)^h \|x_0^\star - x_0^{\mathrm{p}}\| + L_{w,h}(\epsilon_d + \tilde{\eta}),\end{aligned} \tag{50}$$

where the first inequality follows using Lemma 10. Analogously, we have

$$\begin{aligned}\|x_{h+1}^\star - x_{h+1}^{\mathrm{p},l}\| &\leq (L + L_w)^h \|x_0^\star - x_0^{\mathrm{p},l}\| + L_{w,h}(\epsilon_d + \tilde{\eta}) \\ &\leq (L + L_w)^h \sum_{i=0}^{\delta h - 1} L^i(\epsilon_c + \tilde{\eta}) + L_{w,h}(\epsilon_d + \tilde{\eta}).\end{aligned} \tag{51}$$

Together, Eqs. (50) and (51) on substituting in Eq. (49) imply

$$\forall h \in \mathbb{N}_{[0,\mathrm{H}-1]},\ \|x^{\mathrm{p}}_{h+1} - x^{\mathrm{p},l}_{h+1}\| \leq (L + L_w)^{\mathrm{H}} \sum_{i=0}^{\delta h-1} L^i(\epsilon_c + \tilde{\eta}) + 2L_{w,\mathrm{H}}(\epsilon_d + \tilde{\eta}) \leq \epsilon'.$$

and for the input constraints, we have

$$\forall h \in \mathbb{N}_{[0,\mathrm{H}-1]}, \|\pi_h(x^{\mathrm{p}}_h) - \pi_h(x^{\mathrm{p},l}_h)\| \leq L_\pi \|x^{\mathrm{p}}_h - x^{\mathrm{p},l}_h\| \leq \epsilon'.$$

with

$$\epsilon' \geq \max\{1, L_\pi\} \left(\max\{(L + L_w)^{\mathrm{H}}, 1\} L_{\Delta\mathrm{H}}(\epsilon_c + \tilde{\eta}) + 2L_{w,\mathrm{H}}(\epsilon_d + \tilde{\eta})\right). \tag{52}$$

□

Lemma 12. *Let Assumptions 1 to 4 hold. Suppose Algorithm 2 terminates in $\bar{n} \leq n^\star$ iterations with $n^\star$ as defined in Theorem 1. Then it holds that $\Pi^{\mathrm{o},\epsilon}_{\bar{n}}(\mathbb{X}_{\bar{n}}; \mathrm{H}) \subseteq \Pi^{\mathrm{p},\epsilon'}_{\bar{n}}(\mathbb{X}_{\bar{n}}; \mathrm{H})$.*

Proof. Given the sampling rule in Algorithm 2 and using Lemma 2, we know $\exists \bar{n} \leq n : w_{\bar{n}}(x_h, \pi^{\mathrm{p}}(x_h)) < \epsilon_d, \forall h \in \mathbb{N}_{[0,\mathrm{H}_c-1]}, \boldsymbol{f} \in \mathcal{F}_{\bar{n}}, \pi^{\mathrm{p}} \in \Pi^{\mathrm{p}}_{\bar{n}}(x_s, \mathrm{H}_c)$. Note that for all $\delta h \in \mathbb{N}_{[0,\Delta\mathrm{H}]}, \pi \in \Pi^{\mathrm{p}}_{\bar{n}}(x_s; \mathrm{H}+\delta h)$, $\exists \pi_f \in \Pi_{\Delta\mathrm{H}-\delta h} : [\pi, \pi_f] \in \Pi^{\mathrm{p}}_{\bar{n}}(x_s; \mathrm{H}_c)$ using the invariance property of Assumption 3. Hence, $w_{\bar{n}}(x_h, \pi^{\mathrm{p}}(x_h)) < \epsilon_d, \forall h \in \mathbb{N}_{[0,\mathrm{H}+\delta h-1]}, \boldsymbol{f} \in \mathcal{F}_{\bar{n}}, \pi^{\mathrm{p}} \in \Pi^{\mathrm{p}}_{\bar{n}}(x_s, \mathrm{H} + \delta h)$. Using Lemma 11, this implies

$$w_{\bar{n}}(x_h, \pi(x_h)) < \epsilon_d, \forall h \in \mathbb{N}_{[0,\mathrm{H}-1]}, \boldsymbol{f} \in \mathcal{F}_n, \pi \in \Pi^{\mathrm{p},\epsilon'}_{\bar{n}}(\mathbb{X}_n, \mathrm{H}). \tag{53}$$

Using the above statement, the proof follows similar to Lemma 4, where we use

$$\epsilon - \epsilon' > \max\{1, L_\pi\} 2 \sum_{j=0}^{\mathrm{H}-1} (L_w + L)^j (\epsilon_d + L_w L_{\mathrm{H}-j-1} \tilde{\eta}), \tag{54}$$

see also Eq. (27) in Lemma 3. □

Lemma 13 (Guaranteed termination)**.** *Assumptions 1 to 4 hold. Consider $n^\star$ as defined in Theorem 1. Algorithm 2 terminates in $\bar{n} \leq n^\star$ iterations.*

Proof. Analogous to Proposition 2, Algorithm 2 ensures that with every iteration, at least one informative measurement is collected, i.e., $|\mathcal{D}_c| \geq 1$. Following Lemma 2, this implies that there exists $\bar{n} \leq n^\star$ such that $w_{\bar{n},i}(x_h, u_h) < \epsilon_d, \forall i \in \mathbb{N}_{[1,n_x]}, h \in \mathbb{N}_{[0,\mathrm{H}_c-1]}, \boldsymbol{f} \in \mathcal{F}_n, \pi \in \Pi^{\mathrm{p}}_{\bar{n}}(x(k), \mathrm{H}_c)$ where $x_{h+1} = \boldsymbol{f}(x_h, u_h), u_h = \pi_h(x_h)$ and $x_0 = x(k)$.

In the following, we show that this ensures satisfaction of the termination criteria (35). From Lemma 12, we know that $\pi^{\mathrm{o}} \in \Pi^{\mathrm{o},\epsilon}_{\bar{n}}(\mathbb{X}_n; \mathrm{H}) \subseteq \Pi^{\mathrm{p},\epsilon'}_{\bar{n}}(\mathbb{X}_n; \mathrm{H})$. Since π^{o} and x^{o} are feasible candidate for pessimistic problem (10) we get,

$$J^{\mathrm{p}}(x^{\mathrm{p}}, \boldsymbol{\mu}_n; \pi^{\mathrm{p}}) \geq J^{\mathrm{p}}(x^{\mathrm{o}}, \boldsymbol{\mu}_n; \pi^{\mathrm{o}}) \tag{55}$$

$$\overset{\text{(i)}}{=} J(x^{\mathrm{o}}, \boldsymbol{\mu}_n; \pi^{\mathrm{o}}) - \mathbb{E}_\eta \left[L_r \sum_{h=0}^{\mathrm{H}-1} \sum_{i=0}^{h-1} L^i w_{n-1}(x_i, \pi^{\mathrm{o}}_i(x_i)) \right] \tag{56}$$

$$\overset{\text{(ii)}}{\geq} J(x^{\mathrm{o}}, \boldsymbol{\mu}_n; \pi^{\mathrm{o}}) - \mathbb{E}_\eta \left[L_r \sum_{h=0}^{\mathrm{H}-1} \sum_{i=0}^{h-1} L^i (L_w \|x^\star_i - x_i\| + \epsilon_d) \right] \tag{57}$$

$$\overset{\text{(iii)}}{\geq} J(x^{\mathrm{o}}, \boldsymbol{\mu}_n; \pi^{\mathrm{o}}) - \mathbb{E}_\eta \left[L_r \sum_{h=0}^{\mathrm{H}-1} \sum_{i=0}^{h-1} L^i (L_w L_{w,i}(\epsilon_d + \tilde{\eta}) + \epsilon_d) \right] \tag{58}$$

$$\geq J(x^{\mathrm{o}}, \boldsymbol{f}^{\mathrm{o}}; \pi^{\mathrm{o}}) - L_r \sum_{h=0}^{\mathrm{H}-1} \sum_{i=0}^{h-1} L^i (L_w L_{w,i}(\epsilon_d + \tilde{\eta}) + \epsilon_d) - 2L_r \sum_{h=0}^{\mathrm{H}-1} L_{w,h}(\epsilon_d + \tilde{\eta}) \tag{59}$$

Here (i) follows from the definition of pessimistic objective (10). x_i, and $x_i^\star$ denote the trajectory under with mean and true dynamics while x_i is perturbed noise whereas $x_h^\star$ is noise-free and $w_{\bar{n}}(x_i^\star, \pi_i^{\mathrm{o}}(x_i^\star)) < \epsilon_d$. Step (iii) follows from the following. Consider,

$$
\begin{aligned}
\|x_{h+1}^\star - x_{h+1}\| &\leq \|\boldsymbol{f}^\star(x_h^\star, \pi_h^{\mathrm{o}}(x_h^\star)) - \boldsymbol{\mu}_{\bar{n}}(x_h, \pi_h^{\mathrm{o}}(x_h)) - \eta\| && (60)\\
&\leq \tilde{\eta} + \|\boldsymbol{f}^\star(x_h^\star, \pi_h^{\mathrm{o}}(x_h^\star)) - \boldsymbol{f}^\star(x_h, \pi_h^{\mathrm{o}}(x_h)) + \boldsymbol{f}^\star(x_h, \pi_h^{\mathrm{o}}(x_h)) - \boldsymbol{\mu}_{\bar{n}}(x_h, \pi_h^{\mathrm{o}}(x_h))\|\\
&\leq (L + L_w)\|x_h^\star - x_h\| + \epsilon_d + \tilde{\eta} && (61)\\
&\leq L_{w,h+1}(\epsilon_d + \tilde{\eta}) && (62)
\end{aligned}
$$

The step in (61) follows using $w_{\bar{n}}(x_h^{\mathrm{o}}, \pi_h^o(x_h^{\mathrm{o}})) < L_w\|x_h^\star - x_h^{\mathrm{o}}\| + w_{\bar{n}}(x_h^\star, \pi_h^o(x_h^\star))$ and $w_{\bar{n}}(x_h^\star, \pi_h^o(x_h^\star)) < \epsilon_d$.

Finally, the last inequality in (59) follows from the following,

$$
\begin{aligned}
|J(x^{\mathrm{o}}, \boldsymbol{\mu}_n; \pi^{\mathrm{o}}) - J(x^{\mathrm{o}}, \boldsymbol{f}^o; \pi^{\mathrm{o}})| &\leq \sum_{h=0}^{\mathrm{H}-1} |r(x_h, \pi^{\mathrm{o}}(x_h)) - r(x_h^{\mathrm{o}}, \pi^{\mathrm{o}}(x_h^{\mathrm{o}}))| && (63)\\
&\leq L_r \sum_{h=0}^{\mathrm{H}-1} \|x_h - x_h^{\mathrm{o}}\| && (64)\\
&\leq L_r \sum_{h=0}^{\mathrm{H}-1} \|x_h - x_h^\star\| + \|x_h^\star - x_h^{\mathrm{o}}\| && (65)
\end{aligned}
$$

where x_h^{o} denotes the trajectory under optimistic dynamics, perturbed by the same noise sequence as x_h. The trajectory difference bound from (62) also holds for the optimistic dynamics, since $\boldsymbol{f}^{\mathrm{o}} \in \mathcal{F}_{\bar{n}}$. Hence, on substituting (62) in (65), we get,

$$
|J(x^{\mathrm{o}}, \boldsymbol{\mu}_n; \pi^{\mathrm{o}}) - J(x^{\mathrm{o}}, \boldsymbol{f}^o; \pi^{\mathrm{o}})| \leq 2L_r \sum_{h=0}^{\mathrm{H}-1} L_{w,h}(\epsilon_d + \tilde{\eta}). \qquad (66)
$$

□

Theorem 2. *Let Assumptions 1 to 5 hold. Consider $n^\star$ as in Theorem 1. With probability atleast* $1-\delta$,

- *All the optimization problems in Algorithm 2 are feasible for $\forall n \geq 0$;*
- *Algorithm 2 guarantees safety for the unknown system at all times: $x(k) \in \mathcal{X}, u(k) \in \mathcal{U}, \forall k \in \mathbb{N}$;*
- *Algorithm 2 is guaranteed to terminate in at most $n^\star$ iterations;*
- *Algorithm 2 returns policy π^{r} that satisfies Objective 2 i.e.,achieves close-to-optimal performance.*

Proof. Feasibility: First, we prove the feasibility of Algorithm 2. The control invariance property of Assumption 3 ensures that $[\pi_{\mathrm{f}}, \ldots, \pi_{\mathrm{f}}] \in \Pi_0^{\mathrm{p},\epsilon'}(\mathbb{X}_0, \mathrm{H})$. Since the $\Pi_0^{\mathrm{p},\epsilon'}(\mathbb{X}_0, \mathrm{H})$ is not empty, this guarantees feasibility of Problem (10) (right) in Line 4. The feasibility of the optimistic Problem (10) (left) is similarly ensured by the control invariance property with tightening of the safe set with ϵ.

Feasibility of Problem (12) in Line 8 at any $n \geq 1$ is ensured using standard MPC arguments for recursive feasibility [60]. Without loss of generality, let π^{old} be a feasible policy at any n and $h' \in \mathbb{N}_{[0,\mathrm{H}_c-1]}$ be the step where we obtain an informative location ($|\mathcal{D}_c| \geq 1$) under π^{old}. We build a feasible candidate sequence at iteration $n+1$ with $\pi \in \Pi_{\mathrm{H}_c}$ and $x_h \in \mathcal{X}$ be state under any $\boldsymbol{f} \in \mathcal{F}_{n+1}$, by shifting the previous feasible solution $\pi_h = \pi_{h'+h}^{\mathrm{old}}, \forall h \in \mathbb{N}_{[0,\mathrm{H}_c-h'-1]}$ and appending $\pi_h = \pi_f, \forall h \in \mathbb{N}_{[\mathrm{H}_c-h',\mathrm{H}_c-1]}$, from Assumption 3. Feasibility follows from $x_h \in \mathcal{X}, \forall h \in \mathbb{N}_{[0,\mathrm{H}_c-h'-1]}$ where x_h is propagated under any $\boldsymbol{f} \in \mathcal{F}_{n+1} \subseteq \mathcal{F}_n$ (nestedness Eq. (4)) with the previous feasible policy; and $x_h \in \mathbb{X}_n \subseteq \mathbb{X}_{n+1} \subseteq \mathcal{X}, \forall h \in \mathbb{N}_{[\mathrm{H}_c-h',\mathrm{H}_c-1]}$ respectively due to "invariance" and "monotonicity" of the safe set $\mathbb{X}_n$ (Assumption 3). Moreover, note that the constraint $w_n(x_h, u_h) \geq \epsilon_d - \nu$ in Problem (12) is always feasible by choosing ν sufficiently large.

At $k = 0$, $\Pi_0^{\mathrm{p}}(x(k); \mathrm{H}_c)$ is non-empty due to the control invariance property and $x(0) \in \mathbb{X}_0$. From then on, for any further $n \geq 1$, the solution of $n-1$ is still feasible since $\mathbb{X}_n \subseteq \mathbb{X}_{n+1}$ by Assumption 3, $\mathcal{F}_n \supseteq \mathcal{F}_{n+1}$ by construction (4) and the control invariance property ensures that the system can stay in the safe set. In Line 13, Problem (12) is feasible with $\nu = 0$. Otherwise it would hold that $w_{\bar{n}}(x_h, u_h) < \epsilon_d, \quad \forall h \in \mathbb{N}_{[0,\mathrm{H}_c-1]}, \boldsymbol{f} \in \mathcal{F}_{\bar{n}}, \pi^{\mathrm{p}} \in \Pi^{\mathrm{p}}_{\mathrm{H}_c}(x_s)$, with $x_{h+1} = \boldsymbol{f}(x_h, u_h), u_h = \pi_h(x_h)$, which implies $\Pi_n^{\mathrm{o},\epsilon}(\mathbb{X}_n; \mathrm{H}) \subseteq \Pi_n^{\mathrm{p},\epsilon'}(\mathbb{X}_n; \mathrm{H})$ by Lemmas 11 and 12. This would already ensure that the termination condition of Algorithm 2 is satisfied and thus the algorithm will reach Line 13 only if it is feasible. Hence, all optimization problems are feasible in Algorithm 2 for all $n \geq 0$.

Safety: Analogous to Theorem 1, safety is ensured since Algorithm 2 applies the policy $\pi^{\mathrm{p}} \in \Pi_n^{\mathrm{p}}(x(k); \mathrm{H}_c), \forall n \geq 0$ which, by definition, ensures constraint satisfaction $\forall \boldsymbol{f} \in \mathcal{F}_n$. By Assumption 1 and Lemma 1, since the unknown system $\boldsymbol{f}^\star \in \mathcal{F}_n$, thereby guaranteeing constraint satisfaction for the unknown system (1) as well.

Finite termination: Lemma 13 ensures that the defined termination criterion is satisfied in $\bar{n} \leq n^\star$ iterations.

Close to optimal performance: First, we show that the agent satisfies constraints after following the returned policy $[\hat{\pi}, \pi^{\mathrm{p}}]$ in Lemma 8. Then Lemma 9 ensures that once the termination criterion is satisfied (early termination not necessarily uniformly reducing uncertainty), then the returned policy satisfies Objective 2. □

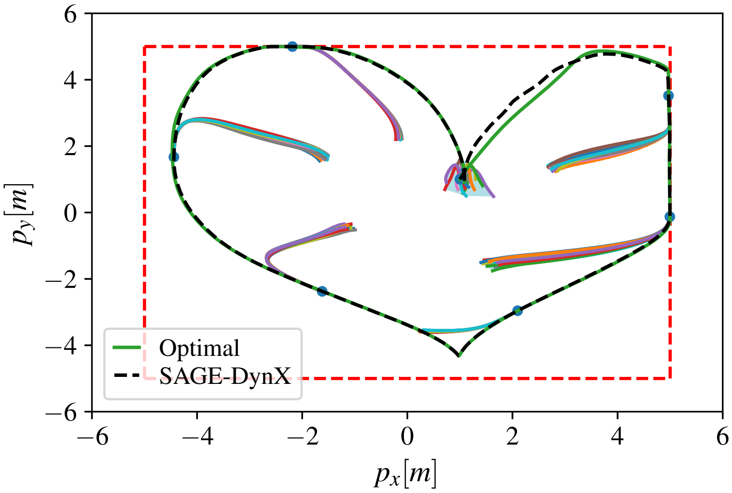


Figure 8: Demonstration of the SageDynX algorithm in the drone navigation environment. The drone begins at the initial position (1,1) and aims to follow a heart-shaped reference trajectory while satisfying state constraints indicated by dashed red lines. The green line represents the optimal trajectory computed by a clairvoyant agent (known system dynamics). The dashed black line shows the agent's trajectory using SageDynX. The thin multicolored lines illustrate predicted trajectories using different sampled dynamics starting from the drone's position at different time steps, highlighted by cyan dots. SageDynX initially deviates from the optimal behavior, gathers informative data, and quickly converges towards close to optimal performance.

D Experiment

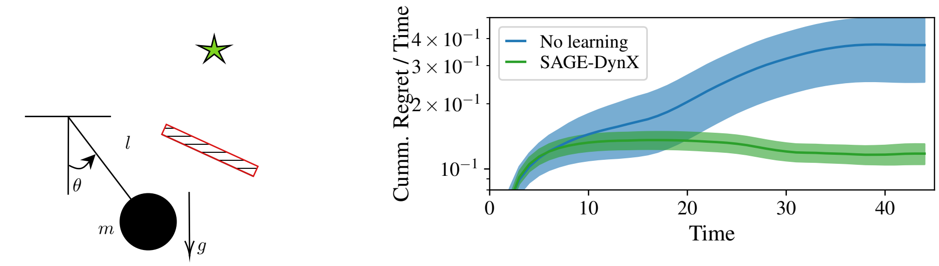
We first discuss implementation details, and later explain each of the environment models, which include an inverted pendulum, drone navigation, and car racing.

Implementation details. Across different environments, we employ Gaussian processes or a distribution over a finite number of basis functions as probabilistic models (i.e., Gaussian processes with degenerate kernels) to capture the unknown system dynamics. To obtain coverage over the dynamics set, we sample multiple dynamics functions from the GP posterior similar to [63], as discussed in Remark 3. For solving the resulting non-linear finite-horizon optimization problem with sampled dynamics, we utilize the Sequential Quadratic Programming (SQP) algorithm proposed in [63]. This optimization procedure yields a unique control input sequence that guarantees safety across all sampled dynamics models. According to [22], a finite number of GP samples and an appropriate tolerance yield a reliable over-approximation of the reachable set induced by the dynamics set $\mathcal{F}_n$ and thus ensure constraint satisfaction. This requires constraint tightening based on the number of samples. In experiments, we ignore the tightening and directly calibrate the number of samples such that it provides a good approximation of the reachable set and ensures constraint satisfaction. Moreover, we directly optimize the open-loop action sequence instead of affine policies (Remark 3). In the following, we provide details specific to each environment used in our experiments:

Inverted Pendulum. The pendulum with unknown dynamics, where the ground-truth evolution (available only to a clairvoyant agent) is described by the non-linear model:

$$\begin{bmatrix}\theta(k+1)\\ \\ \omega(k+1)\end{bmatrix} = \begin{bmatrix}\theta(k)+\omega(k)\Delta+\eta_1(k)\\ \\ \omega(k)-g_a\sin(\theta(k))\Delta/l+\alpha(k)\Delta+\eta_2(k)\end{bmatrix}.$$

The state vector is $x=[\theta,\omega]^\top$, where $\theta[\mathrm{rad}]$ is the angular position, $\omega[\mathrm{rad/s}]$ is the angular velocity, and the control input is the angular acceleration $\alpha[\mathrm{rad/s}^2]$. The dynamics are stochastic, corrupted with uniform noise, $[\eta_1,\eta_2]^\top$ bounded by $\tilde{\eta}=10^{-3}$. The pendulum has a length of $l=1\mathrm{m}$, the system is discretized with $\Delta=0.015\mathrm{s}$, and the acceleration due to gravity is $g_a=9.81\mathrm{m/s}^2$. The GP is trained using $|\bar{\mathcal{D}}_0|=27$ prior data points on an equally-spaced $3\times3\times3$ mesh grid in the constraint set $\mathcal{X}\times\mathcal{U}=\{(\theta,\omega,\alpha)\in[-2.14,1.14]\times[-2.5,2.5]\times[-8,8]\}$. We use a squared exponential kernel with its hyperparameters optimized using the maximum likelihood estimate. The task is to



(a) Inverted pendulum setup

(b) Regret comparison in pendulum

Figure 9: Illustration of the pendulum example. Fig. 9a: The pendulum begins at the bottom position and aims to reach closer to the green star, while being constrained by the physical red wall and a limit on the angular velocity. Fig. 9b shows the cumulative regret over time (averaged across runs). SAGEDYNX has significantly lower regret as compared to the no-learning baseline, implying that it more closely follows the optimal trajectory of the clairvoyant agent.

control the pendulum from $x = (0, 0)$ to a desired state of x_{des} marked by a green start in Fig. 9a. The reward function is given by $r(\theta_h, \alpha_h) = 50(x_h - x_{\text{des}})^2 + 0.1\alpha_h^2, \forall h \in \mathbb{N}_{[0,\mathrm{H}_c-1]}$, where $\mathrm{H}_c = 31$.

To ensure a good coverage of the dynamics set $\mathcal{F}$, we used 50 dynamics functions sampled from the GP. We construct a safe set (Assumption 3) around the origin by computing a common Lyapunov function using Jacobians (around the origin) of the dynamics sampled from the set $\mathcal{F}_n$, as done in [22]. We recursively solve a pessimistic problem, which finds a unique common control sequence that keeps all the dynamics safe and returns them to the safe set. In line with SAGEDYNX, instead of executing the entire plan (return back), we replan after collecting every measurement, which are collected after every five time steps.

Fig. 9b compares the performance SAGEDYNX with a no-learning algorithm, which does not actively move to informative states and does not incorporate online measurements to update the model. We plot cumulative regret over time, where regret is computed with the position difference from the clairvoyant agent at any time step. The no-learning baseline, due to the high model uncertainty, avoids using high angular velocity (to be safe) and deviates significantly from the optimal behaviour. In contrast, SAGEDYNX gathers informative data and then progressively aligns more closely to the optimal behaviour.

Drone Navigation. We consider a nonlinear drone system described by the following continuous-time dynamics [42, 43]:

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{\phi} \\ \dot{v}_x \\ \dot{v}_y \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} v_x \cos(\phi) - v_y \sin(\phi) \\ v_x \sin(\phi) + v_y \cos(\phi) \\ \dot{\phi} \\ v_y \dot{\phi} - g \sin(\phi) \\ -v_x \dot{\phi} - g \cos(\phi) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{m} & \frac{1}{m} \\ \frac{l}{J} & -\frac{l}{J} \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \cos(\phi) \\ -\sin(\phi) \\ 0 \end{bmatrix} d$$

In this model, p_x and p_y denote the drone's horizontal and vertical positions, and v_x, v_y represent the corresponding velocities expressed in the body-fixed frame. The variables ϕ and $\dot{\phi}$ refer to the drone's orientation angle and its angular velocity. The control vector $\mathbf{u} = [u_1, u_2]^\top$ consists of the thrust forces generated by the two propellers. The constant $d = 0.1$ accounts for external wind disturbances. Remaining constants $g = 9.81$ $[\mathrm{m/s^2}]$, $l = 0.25$ $[\mathrm{m}]$, $J = 0.00383$ and $m = 1$ $[\mathrm{kg}]$ denote gravity, the distance from each propeller to the vehicle's center, moment and mass of the drone, respectively. We discretize the continuous-time drone dynamics described above using the Euler discretization method with the time step of $\Delta = 0.1$ $[\mathrm{s}]$.

We model the dynamics using a finite set of basis functions, incorporating a total of 21 features with unknown parameters. To initialize the model, we use 2 prior data points along each state and input dimension as a mesh grid over the constraint set given by $\mathcal{X} \times \mathcal{U} = \{(x, u) \in [-5, 5] \times [-5, 5] \times$

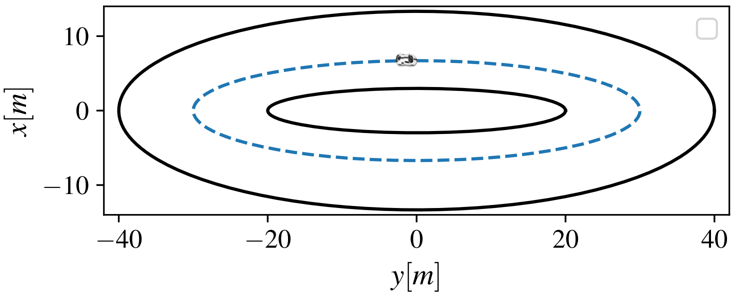


Figure 10: Illustration of the car track depicted by the black line. The car is required to follow the reference trajectory (blue dashed line) while respecting state and input constraints, as well as staying within the track boundaries.

$[-\pi, \pi] \times [-2, 2] \times [-2, 2] \times [-1, 1] \times [-1, 5] \times [-1, 5]\}$. The drone must learn the dynamics online while staying within these constraints. As shown in Fig. 8, the drone starts at $(p_x, p_y) = (1, 1)$ and the task requires tracking of a heart-shaped reference, which is incorporated using a time-varying reward function. The exact equation for generating a heart shape can be found in the submitted code. The entire reference is divided into 500 discrete steps, and the agent optimizes over rewards with a receding horizon (rewards based on a moving reference) of length $\mathrm{H}_c = 31$ that advances one step per action. The pessimistic planning uses 15 samples from the dynamic functions based on the updated model set $\mathcal{F}_n$. The safe set $\mathbb{X}_n$ is an ellipsoid centered around any state with velocity zero, ensuring that the pessimistic plan ends with low velocity. We do not observe any constraint violations in the experiment. We replan after each measurement, which is collected at every alternate time step.

Car Racing. We model the nonlinear car dynamics using a kinematic bicycle model [44] as follows:

$$\begin{bmatrix} x_p(k+1) \\ y_p(k+1) \\ \theta(k+1) \\ v(k+1) \end{bmatrix} = \begin{bmatrix} x_p(k) + v(k)\cos(\theta(k) + \zeta_k)\Delta \\ y_p(k) + v(k)\sin(\theta(k) + \zeta_k)\Delta \\ \theta(k) + v(k)\sin(\zeta_k) l_r^{-1}\Delta \\ v(k) + a(k)\Delta - cv^2(k)\Delta \end{bmatrix}$$

where $\zeta_k = \tan^{-1}\left(\frac{l_r}{l_f + l_r}\tan(\delta(k))\right)$ denotes the slip angle $[\mathrm{rad}]$. The state vector is defined as $x = [x_p, y_p, \theta, v]^\top$, where $[x_p, y_p]^\top$ specifies the vehicle's position in a 2D Cartesian frame $[\mathrm{m}]$, θ $[\mathrm{rad}]$ represents the vehicle's heading angle, and v $[\mathrm{m/s}]$ is the forward velocity. The control vector is given by $u = [\delta, a]^\top$, where δ $[\mathrm{rad}]$ is the steering input and a $[\mathrm{m/s^2}]$ is the applied longitudinal acceleration. The term $cv^2(k)$ denotes the drag force with constant $c = 0.4167$ and the system is discretized with $\Delta = 0.06$ $[\mathrm{s}]$. The parameters $l_f = 1.105$ $[\mathrm{m}]$ and $l_r = 1.738$ $[\mathrm{m}]$ represent the distances from the vehicle's center of mass to the front and rear axles, respectively.

We model vehicle dynamics using a finite set of basis functions, incorporating a total of nine features. To initialize the model, we use 64 prior data points, generated by sampling two values along each state and input dimension. These data points are concentrated near the initial state to provide a reliable prior. The car must learn the dynamics online while staying within track constraints defined by two elliptical boundaries centered at $(x_e, y_e) = (0, 0)$. The outer ellipse is given by $(x_p - x_e)^2 + 9(y_p - y_e)^2 \leq 1600$ and the inner ellipse by $(x_p - x_e)^2 + 45(y_p - y_e)^2 \geq 400$, forming a corridor the vehicle must remain within as shown in Fig. 10. The reward function is time-varying, with the track divided into 500 discrete steps. The agent optimizes over rewards with a receding horizon of length $\mathrm{H}_c = 31$ that advances one step per action (receding). The reference follows an elliptical path defined by $(x_p - x_e)^2 + 30(x_p - y_e)^2 = 900$, which approximately represents the center line. In addition to the track boundaries, the car is subjected to state constraints and input constraints $\mathcal{X} \times \mathcal{U} = \{(x, u) \in [-45, 45] \times [-15, 15] \times [-40.14, 40.14] \times [-15, 20] \times [-0.6, 0.6] \times [-2, 20]\}$. The pessimistic planning uses 15 samples from the dynamic functions based on the updated model set $\mathcal{F}_n$. We do not observe any constraint violations in the experiment. We replan after each measurement, which is collected at each time step.