Safe Model-Based Meta-Reinforcement Learning: A Sequential Exploration-Exploitation Framework

Thomas Lew, Apoorva Sharma, James Harrison, Marco Pavone
Stanford University, Stanford, CA
{thomas.lew,apoorva,jharrison,pavone}@stanford.edu

Abstract: Safe deployment of autonomous robots in diverse environments requires agents that are capable of safe and efficient adaptation to new scenarios. Indeed, achieving both data efficiency and well-calibrated safety has been a central problem in robotic learning and adaptive control due in part to the tension between these objectives. In this work, we develop a framework for probabilistically safe operation with uncertain dynamics. This framework relies on Bayesian meta-learning for efficient inference of system dynamics with calibrated uncertainty. We leverage the model structure to construct confidence bounds which hold throughout the learning process, and factor this uncertainty into a model-based planning framework. By decomposing the problem of control under uncertainty into discrete exploration and exploitation phases, our framework extends to problems with high initial uncertainty while maintaining probabilistic safety and persistent feasibility guarantees during every phase of operation. We validate our approach on the problem of a nonlinear free flying space robot manipulating a payload in cluttered environments, and show it can safely learn and reach a goal.

Keywords: Safe Learning and Control, Reinforcement Learning, Meta-Learning

1 Introduction

Deployment of truly autonomous robotic systems in changing and unpredictable environments requires agents that are capable of learning during operation and safely adapting to new environments. For example, autonomous robots assisting astronauts in space must be able to identify and safely manipulate complex payloads while avoiding collision with their surroundings. Reinforcement learning (RL) can be an effective approach to controlling uncertain systems [1, 2], and model-based methods in particular enable an agent to consider its uncertainty over dynamics when choosing actions [3]. However, standard model-based reinforcement learning (MBRL) methods do not provide guarantees on maintaining safety during operation. Existing work on safety in MBRL has developed algorithms with strong theoretical guarantees, but has either been limited to linear systems [4], or utilized kernel-based dynamics models which struggle to scale with state dimension, and uncertainty propagation schemes that can be too conservative for practical use [5].

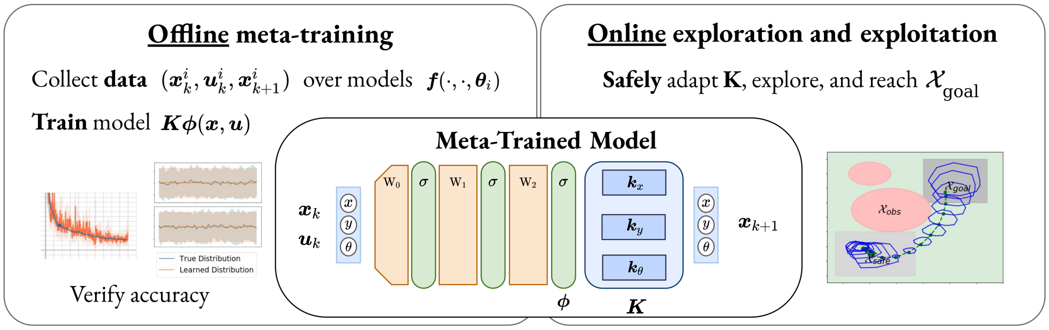

Figure 1: To guarantee safety at all times and reach a goal region $\mathcal{X}_{\text{goal}}$ despite uncertain dynamics $f(\cdot,\cdot,\theta_i)$, our framework consists of an offline phase, where a dataset over multiple models is used to meta-train an uncertain Bayesian model of the system. Then, it is deployed and safely adapts the last layer K of the meta-trained model, autonomously explores the environment to decrease its uncertainty, and safely reaches $\mathcal{X}_{\text{goal}}$.

Preprint. Under review.

In this work, we approach the problem from a complementary angle, leveraging Bayesian meta-learning [6] and sampling-based reachability analysis [7] to develop a framework for safe, nonlinear MBRL that is practically useful and probabilistically safe under assumptions on the quality of meta-learning and uncertainty propagation. To handle high levels of initial uncertainty, our approach relies on decoupling online learning to reduce dynamics uncertainty (the exploration phase) and executing the desired task (the exploitation phase). Our framework autonomously explores until it can safely exploit to carry out the task, and maintains probabilistic safety throughout.

Contributions: This paper proposes a new approach summarized in Figure 1, which addresses the problem of safely reaching a goal in spite of unknown dynamics. Our core contributions are:

- We develop a practical framework for the safe control of uncertain nonlinear dynamical systems, capable of safely performing tasks for highly uncertain systems by autonomously exploring to reduce uncertainty before performing the task.
- We prove probabilistic safety guarantees for our framework, given conditions on the quality of the meta-learning and uncertainty propagation. We discuss how future work may address the problem of formally verifying these conditions and handling cases where they are not met.
- We validate our approach on a challenging, nonlinear, highly uncertain system, and show that we are able to autonomously and safely infer dynamics and reach a goal.

2 Problem Formulation: Safe Navigation to a Goal

The goal of this work is to enable robots to safely navigate from an initial state $\boldsymbol{x}(0)$ to a goal region $\mathcal{X}_{\text{goal}}$ despite highly uncertain dynamics, while minimizing a chosen cost metric $l(\cdot)$ (e.g., fuel consumption). We write the state of the agent at time $k \in \mathbb{N}$ as $\boldsymbol{x}_k \in \mathbb{R}^n$, and $\boldsymbol{u}_k \in \mathbb{R}^m$ denotes the control input. The system follows dynamics $\boldsymbol{x}_{k+1} = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}) + \boldsymbol{\epsilon}_k$, where $\boldsymbol{h}(\cdot)$ is known and $\boldsymbol{g}$ is unknown, and with parameters $\boldsymbol{\theta}$ and stochastic disturbances $\boldsymbol{\epsilon}_k$. We assume the (unobserved) parameter is sampled $\boldsymbol{\theta}_j \sim p(\boldsymbol{\theta})$ at the beginning of each episode j, and fixed for the duration of the episode. These parameters correspond to unknown features that vary between episodes, e.g. the mass and inertia of a payload. We assume that the $\boldsymbol{\epsilon}_k = (\epsilon_k^1, \ldots, \epsilon_k^n)$ are uncorrelated over $k \in \mathbb{N}$ and $i = 1, \ldots, n$, and that each ϵ_k^i is σ_{ϵ_i}-subgaussian and bounded ($\epsilon_k^i \in \mathcal{E}_i$).

Critically, this algorithm should guarantee safety *at all times* by respecting system constraints ($\boldsymbol{x}_k \in \mathcal{X}$, $\boldsymbol{u}_k \in \mathcal{U}$, where $\mathcal{X}, \mathcal{U}$ are feasible state and control spaces) and avoiding obstacles ($\boldsymbol{x}_k \notin \mathcal{X}_{\text{obs}}$, where $\mathcal{X}_{\text{obs}}$ is the set of obstacles). Due to the stochasticity of the system and the uncertain dynamics, strictly enforcing all constraints for all times may be challenging without further assumptions, e.g., bounded model mismatch. Instead, we enforce all constraints with a single chance constraint at probability level $(1-\delta) \in (0,1)$. Specifically, we define a joint chance constraint which should hold at all times until the goal $\mathcal{X}_{\text{goal}}$ is reached, and the problem is expressed as

Chance-Constrained Optimal Control Problem (CC-OCP)

$$\min_{\boldsymbol{x}_{0:N}, \boldsymbol{u}_{0:N-1}} \quad \mathbb{E}\Bigg(\sum_{k=0}^{N} l(\boldsymbol{x}_k, \boldsymbol{u}_k) \Bigg) \tag{1a}$$

$$\text{subject to} \quad \boldsymbol{x}_{k+1} = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}) + \boldsymbol{\epsilon}_k, \;\; \boldsymbol{x}_0 = \boldsymbol{x}(0), \quad k = 0, \ldots, N-1, \tag{1b}$$

$$\mathbb{P}\Bigg(\bigwedge_{k=1}^{N} \big(\boldsymbol{x}_k \in \mathcal{X}_{\text{free}}\big) \;\cap\; \bigwedge_{k=0}^{N-1} \big(\boldsymbol{u}_k \in \mathcal{U}\big) \;\cap\; \big(\boldsymbol{x}_N \in \mathcal{X}_{\text{goal}}\big) \Bigg) \geq (1-\delta). \tag{1c}$$

where $\mathcal{X}_{\text{free}} = \mathcal{X} \setminus \mathcal{X}_{\text{obs}}$, and $\mathcal{X}_{\text{goal}} \subset \mathcal{X}_{\text{free}}$ and N is the total duration of the problem (possibly infinite). Note that this problem formulation can be equivalently described as a constrained Markov decision process with a continuous state and action space, general nonlinear stochastic dynamics, and a non-convex cost function. Satisfying safety constraints with unknown dynamics at all times is extremely difficult without further information; our approach relies on the following non-standard (but practically realistic) assumptions.

Assumption 1. We have access to a dataset of trajectories $\{\{(\boldsymbol{x}_k^j, \boldsymbol{u}_k^j, \boldsymbol{x}_{k+1}^j)\}_{k=0}^{N_j}\}_{j=1}^{D}$, where each trajectory $\{(\boldsymbol{x}_k^j, \boldsymbol{u}_k^j, \boldsymbol{x}_{k+1}^j)\}_{k=0}^{N_j}$ is obtained from the true dynamics with $\boldsymbol{g}(\cdot,\cdot,\boldsymbol{\theta}_j) + \boldsymbol{\epsilon}$, and $\boldsymbol{\theta}_j \sim p(\boldsymbol{\theta})$.

Assumption 2. We assume $\boldsymbol{x}(0) \in \mathcal{X}_0 \subset \mathcal{X}_{\text{free}}$, where $\mathcal{X}_0$ is a control invariant set and we have a feedback controller $\boldsymbol{\pi}(\cdot) : \mathcal{X}_0 \to \mathcal{U}$ under which it is possible to remain in $\mathcal{X}_0$ for all $\boldsymbol{\theta}$ and $\boldsymbol{\epsilon}$ [5].

The first assumption reflects that we have access to prior information about plausible trajectories, generated from sampled parameters $\boldsymbol{\theta}_j$, $j=1,\dots,D$. Such information may come from, for example, previous operation of a robot in similar environments, or data generated from simulations with different parameters. This information motivates our use of meta-learning to encode this information and characterize the uncertainty over dynamics, as discussed in the next section. The second assumption reflects that the system is initially stable and satisfies all constraints under a nominal controller (e.g., regulated to a stable linearization point using a simple feedback law such as LQR).

Notations: Let $\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$ the multivariate normal distribution of mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, and $\chi^2_d(p)$ the p-th quantile of the χ^2 distribution with d dofs. For any $\mathbf{a},\mathbf{b}\in\mathbb{R}^d$, and $\mathbf{A}$ a $d\times d$ positive definite matrix, define $\|\mathbf{a}\|^2_{\mathbf{A}}=\mathbf{a}^T\mathbf{A}\mathbf{a}$, and $\langle\mathbf{a},\mathbf{b}\rangle_{\mathbf{A}}=\mathbf{a}^T\mathbf{A}\mathbf{b}$.

3 Background: Bayesian Meta-Learning and Confidence Sets

Bayesian Meta-Learning: Our approach leverages a model for the unknown portion of system dynamics $\boldsymbol{g}$ which expresses the uncertainty regarding the true dynamics, and can efficiently update this uncertainty as we observe transitions from the true system. To this end, we employ the Bayesian meta-learning architecture presented in [6, 8], which the authors refer to as ALPaCA. Meta-learning (or "learning-to-learn") [9, 10, 11] aims to train a model capable of rapid adaptation to a distribution of tasks via training on the loss of the posterior model, adapted to a given task[1]. ALPaCA models the unknown dynamics as

$$\hat{\boldsymbol{g}}(\boldsymbol{x},\boldsymbol{u})=\boldsymbol{K}\boldsymbol{\phi}(\boldsymbol{x},\boldsymbol{u}),\tag{2}$$

where $\boldsymbol{\phi}(\cdot,\cdot)$ is a feed-forward neural network with output dimension d, and $\boldsymbol{K}$ is an $n\times d$ matrix which can be thought of as the last layer linear weights. The uncertainty in the space of dynamics functions is encoded through a normal distribution on each row $\boldsymbol{k}_i$ of $\boldsymbol{K}$: $\boldsymbol{k}_i\sim\mathcal{N}(\bar{\boldsymbol{k}}_i,\sigma^2_{\epsilon_i}\boldsymbol{\Lambda}_i^{-1})$.

The linear structure of this model allows for efficient online updates whose behavior is well understood. Given a set of transitions from interaction $\{(\boldsymbol{x}_0,\boldsymbol{u}_0,\boldsymbol{x}_1),\dots,(\boldsymbol{x}_t,\boldsymbol{u}_t,\boldsymbol{x}_{t+1})\}$, we can compute the posterior over each i-th row of $\boldsymbol{K}$ using linear regression

$$\boldsymbol{\Lambda}_{i,t}=\Phi_{t-1}^T\Phi_{t-1}+\boldsymbol{\Lambda}_{i,0},\qquad \bar{\boldsymbol{k}}_{i,t}=\boldsymbol{\Lambda}_{i,t}^{-1}(\Phi_{t-1}^T\boldsymbol{G}_{i,t}+\boldsymbol{\Lambda}_{i,0}\bar{\boldsymbol{k}}_{i,0}),\qquad i=1,\dots,n,\tag{3}$$

$\boldsymbol{G}_t^T=[\boldsymbol{x}_1-\boldsymbol{h}(\boldsymbol{x}_0,\boldsymbol{u}_0),\dots,\boldsymbol{x}_t-\boldsymbol{h}(\boldsymbol{x}_{t-1},\boldsymbol{u}_{t-1})]\in\mathbb{R}^{n\times t}$, and $\Phi_{t-1}^T=[\boldsymbol{\phi}(\boldsymbol{x}_0,\boldsymbol{u}_0),\dots,\boldsymbol{\phi}(\boldsymbol{x}_{t-1},\boldsymbol{u}_{t-1})]\in\mathbb{R}^{d\times t}$.

Offline, this model is meta-trained on a dataset of trajectories corresponding to different system dynamics sampled from the distribution over possible systems. This procedure consists of backpropagating the loss through the posterior predictive distribution to learn the neural network features $\boldsymbol{\phi}$ as well as the parameters $(\bar{\boldsymbol{k}}_{i,0},\boldsymbol{\Lambda}_{i,0})$ of the prior distribution over each row of the last layer $\boldsymbol{K}$; we refer the reader to [6, 8] for more details. In this way, the model encodes the dynamics uncertainty evident in the dataset into a parametric form in a learned feature space, providing a structured uncertainty representation useful for planning.

Probabilistic Confidence Sets: Our approach requires solving a stochastic optimization problem subject to chance constraints, i.e., ensuring that constraints hold with high probability. To do so, we leverage the concept of confidence sets. We say $\mathcal{S}\subset\mathbb{R}^n$ is a confidence set of probability p for the random variable $\boldsymbol{x}\in\mathbb{R}^n$ if $\Pr(\boldsymbol{x}\in\mathcal{S})\geq p$. The construction of confidence sets enables one to consider $\boldsymbol{x}\in\mathcal{S}$ only, and relax the generally intractable chance-constrained stochastic problem in (1). We follow this approach, similar to [13, 14, 5], to transcribe **(CC-OCP)** into a deterministic problem that can be efficiently solved by a general purpose non-convex solver, e.g. leveraging sequential convex programming [14].

Employing this technique with learned, adaptive models introduces several key challenges. First, we must construct confidence sets corresponding to the joint chance constraint over the entire trajectory. These sets must depend on both the chosen action sequence, as well as the current state of the adaptive model. Second, it is critical that the deterministic problem remains feasible, such that the agent is always able to choose actions that are guaranteed to be safe. This is not trivial, as with high uncertainty, tight control constraints, and a long planning horizon, the problem may become infeasible over time.

[1] The exact mechanism of "adaptation" to a task is at the core of the meta-learning algorithm: in MAML [10] it consists of a gradient step; in recurrent models it occurs via the hidden state dynamics [12], and in ALPaCA [6] the update consists of Bayesian linear regression on the last layer.

4 Quantifying Prediction Accuracy for Safe Chance-Constrained Planning

In order to ensure overall safety, i.e. by satisfying the joint chance constraint (1c) for all times until $\mathcal{X}_{\text{goal}}$ is reached, we require confidence *tubes* over trajectories. Indeed, enforcing a chance constraint at each timestep, as in [14, 15, 16, 17, 18], does not guarantee safety of the whole trajectory [19]. Furthermore, we must construct this confidence tube in a manner that accounts for the stochastic (due to disturbances) *online* adaptation process of the learned model. To do so, we require two key steps: (1) quantifying the prediction accuracy of the meta-learned dynamics model through confidence sets over parameters, and (2) performing reachability analysis using these sets to reformulate the joint chance constraint, and obtain a deterministic problem that can be solved via standard optimization tools. In this section, we describe these steps in detail, discuss required assumptions, derive confidence sets, and provide a conservative deterministic reformulation of **(CC-OCP)**.

4.1 Confidence Sets for Learned Parameters and Guarantees

The linear uncertainty representation of the meta-learned dynamics model enables construction of confidence sets over dynamics models that hold throughout the online learning process, by leveraging results from the literature on linear contextual bandits. These finite-sample *online* learning bounds rely on two critical assumptions on the results of the *offline* meta-learning process: (1) that the meta-learning model is capable of fitting the true system dynamics online, and (2) the uncertainty estimates that are meta-learned represent a conservative prior over the true dynamics functions. We formalize both of these assumptions below:

Assumption 3 (Capacity of meta-learned dynamics model)**.** For all possible $\boldsymbol{\theta}$, there exists $\boldsymbol{k}_i^* \in \mathbb{R}^d$ such that $\langle \boldsymbol{k}_i^* \,;\, \boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{u}) \rangle = \boldsymbol{g}_i(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta})$ for all $\boldsymbol{x} \in \mathcal{X}, \boldsymbol{u} \in \mathcal{U}, i = 1, \ldots, n$.

Assumption 4 (Calibration of meta-learned prior)**.** For $\boldsymbol{\theta} \sim p(\boldsymbol{\theta})$, each $i = 1, \ldots, n$, and $\delta_i = \delta/(2n)$, with probability at least $(1 - \delta_i)$, $\|\boldsymbol{k}_i^* - \bar{\boldsymbol{k}}_{i,0}\|^2_{\boldsymbol{\Lambda}_{i,0}} \le \sigma^2_{\epsilon_i} \chi^2_d(1 - \delta_i)$.

These assumptions state that the true dynamics can be represented as a linear combination of finite dimensional nonlinear features, which applies to a plethora of physical dynamical systems [20, 21]. Further, it assumes that the meta-learning model learns appropriate features for such a representation. Formally verifying these assumptions requires making generalization claims on the meta-learning process, perhaps through a PAC-Bayes analysis [22], and is beyond the scope of this paper. If the dataset has adequate coverage of the state and action spaces and the dynamics distribution $p(\boldsymbol{\theta})$, the offline meta-learning procedure proposed in [6] can approach satisfaction of these assumptions. The validity of these assumptions can be empirically verified through predictive performance on a validation dataset, and techniques such as temperature scaling can be used to ensure calibration in a post-hoc manner [23]. Assumption 3 in particular is comparable to asymptotic representation results in Gaussian process-based methods [24]. We believe our combination of meta-learning with these assumptions motivates two directions of future work that we do not address in this paper: safety analysis with feature mismatch, and finite sample guarantees for meta-learning models.

Recall that as we observe transition tuples $\{(\boldsymbol{x}_\tau, \boldsymbol{u}_\tau, \boldsymbol{x}_{\tau+1})\}_{\tau=0}^{t-1}$ from the true system corrupted by process noise, our model updates its information state, encoded in the parameters $\boldsymbol{k}_{i,t}, \boldsymbol{\Lambda}_{i,t}$ for each dimension i, using equations (3). Given this structure, we can define an *information state dependent* confidence set over model parameters that accounts for this adaptation and *holds uniformly over all future timesteps* $t > 0$ with high probability.

Theorem 1 (Uniformly Calibrated Confidence Sets)**.** *Consider the true system* (1b)*, with* σ_ϵ*-subgaussian bounded noise, modeled using the meta-learning model* (2)*, which is sequentially updated with online data from* (1b) *using* (3)*, leading to the posterior parameters* $(\bar{\boldsymbol{k}}_{i,t}, \boldsymbol{\Lambda}_{i,t})$ *for each dimension* i=$1, \ldots, n$*. Assume that Assumptions 3 and 4 hold, and define*

$$\beta_i(\boldsymbol{\Lambda}_{i,t}, \delta_i) = \sigma_{\epsilon_i} \left(\sqrt{2 \log \left(\frac{1}{\delta_i} \frac{\det(\boldsymbol{\Lambda}_{i,t})^{1/2}}{\det(\boldsymbol{\Lambda}_{i,0})^{1/2}} \right)} + \sqrt{\frac{\lambda_{\max}(\boldsymbol{\Lambda}_{i,0})}{\lambda_{\min}(\boldsymbol{\Lambda}_{i,t})} \chi^2_d(1-\delta_i)} \right), \tag{4}$$

$$\textit{and} \qquad \mathcal{C}^\delta_{i,t}(\bar{\boldsymbol{k}}_{i,t}, \boldsymbol{\Lambda}_{i,t}) = \{\boldsymbol{k}_i \mid \left\|\boldsymbol{k}_i - \bar{\boldsymbol{k}}_{i,t}\right\|_{\boldsymbol{\Lambda}_{i,t}} \le \beta_i(\boldsymbol{\Lambda}_{i,t}, \delta_i)\}. \tag{5}$$

$$\textit{Then, for } \delta_i = \delta/(2n), \qquad \mathbb{P}\left(\boldsymbol{k}_i^* \in \mathcal{C}^\delta_{i,t}(\bar{\boldsymbol{k}}_{i,t}, \boldsymbol{\Lambda}_{i,t}) \quad \forall t \ge 0\right) \ge (1 - 2\delta_i). \tag{6}$$

Note that $\lambda_{\max}(\cdot)$, $\lambda_{\max}(\cdot)$ denote the maximum and minimum eigenvalue respectively. The proof of this result and all subsequent results are available in the appendix. We take a frequentist viewpoint, and assume that there exists a fixed $\boldsymbol{k}_i^*$, and that the confidence set $\mathcal{C}_{i,t}^{\delta}$ is a stochastic function of the observed data. By defining the size of $\mathcal{C}_{i,t}^{\delta}$ through time-dependent values of β_i, we can ensure that the event that the random confidence set will exclude the true $\boldsymbol{k}_i^*$ *at any time* $t > 0$ occurs with probability less than (δ/n). This scaling factor is closely related to that used for kernel Gaussian Processes, for which the value of β_i is often too large for practical use and set to a lower value for experiments [25]. Our meta-learning model operates within a finite dimensional feature space, and has β_i values that are practically useable. Furthermore, properties that influence β_i can be regularized during the offline meta-learning process to yield tighter confidence sets, and obtain better performance without compromising safety. We provide further details and results in the appendix.

4.2 Uncertainty-Aware Reachability Analysis and Deterministic Reformulation

In order to reason about how uncertainty in parameters manifests in terms of how the system may behave in the state space (and whether it might violate safety constraints), we must translate confidence sets over parameters into the corresponding, action dependent reachable sets in the state space. Specifically, given a sequence of open-loop control inputs $\boldsymbol{u} = (\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{N-1})$, we define the sequence of reachable sets

$$\mathcal{X}_k^{t,\delta}(\boldsymbol{u}) = \left\{ \boldsymbol{x}_k = \boldsymbol{f}(\cdot, \boldsymbol{u}_{k-1}, \boldsymbol{K}, \boldsymbol{\epsilon}_{k-1}) \circ \ldots \circ \boldsymbol{f}(\boldsymbol{x}_0, \boldsymbol{u}_0, \boldsymbol{K}, \boldsymbol{\epsilon}_0) \;\middle|\; \begin{array}{c} \boldsymbol{x}_0 = \boldsymbol{x}(t),\ \boldsymbol{k}_i \in \mathcal{C}_{i,t}^{\delta},\ \boldsymbol{\epsilon}_j^i \in \mathcal{E}_i, \\ j = 1, \ldots, k-1,\ \ i = 1, \ldots, n \end{array} \right\}, \quad (7)$$

where $k = 1, \ldots, N$, and $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{K}, \boldsymbol{\epsilon}) = \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{K}\boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{\epsilon}$. This definition closely follows [7], for the specific case of a sequence of open-loop control inputs.[2]

Representing uncertainty regarding the system dynamics and the online learning process as confidence sets on the model parameters $\boldsymbol{K}$, and subsequently transforming these sets into reachable sets in the state space enables the relaxation of the original chance-constrained problem by a deterministic one:

Confident Reachability-Aware Optimal Control Problem

$$\min_{\boldsymbol{\mu}, \boldsymbol{u}} \sum_{k=0}^{N} l(\boldsymbol{\mu}_k, \boldsymbol{u}_k), \quad \text{s.t.} \bigwedge_{k=1}^{N} \mathcal{X}_k^{t,\delta} \subset \mathcal{X}_{\text{free}}, \ \bigwedge_{k=0}^{N-1} \boldsymbol{u}_k \in \mathcal{U}, \ \mathcal{X}_N^{t,\delta} \subset \mathcal{X}_{\text{goal}}, \ \mathcal{X}_0^{t,\delta} = \{\boldsymbol{x}(t)\}, \quad (8)$$

where $\boldsymbol{\mu} = (\boldsymbol{\mu}_0, \ldots, \boldsymbol{\mu}_N)$ are the centers of the sequence of reachable sets $\{\mathcal{X}_k^{t,\delta}\}_{k=1}^{N}$, which depend on $(\boldsymbol{\mu}, \boldsymbol{u})$, and satisfy (7). We use a mean-equivalent reformulation of the expected cost[3], as this work is mostly concerned with reaching $\mathcal{X}_{\text{goal}}$ safely, i.e., about ensuring that the joint chance constraint (1c) holds. Further, the cost typically penalizes control inputs, which are deterministic in this work, so this reformulation is reasonable in practice. In the next section, we discuss implementation details to solve this problem and compute these reachable sets.

5 Safe Sequential Exploration-Exploitation Approach and Guarantees

5.1 Approach and Algorithm Overview

The deterministic formulation in (8) enables the use of deterministic trajectory optimization methods to compute a feasible trajectory. However this problem (and **(CC-OCP)**) may be infeasible if the uncertainty in dynamics is too large, requiring a safe exploration framework to reduce uncertainty. Our approach is based on a two-phase approach: when the problem is feasible, we enter the *exploitation* phase; when the problem is infeasible, we instead enter the *exploration* phase. In the exploitation phase, we solve the trajectory optimization problem with the current model uncertainty. In the exploration phase, we instead strictly perform safe exploration, planning an information-gathering trajectory that returns with high probability to the initial safe invariant set. This split yields a tractable sequence of trajectory optimization problems, although it induces sub-optimality relative to the computationally intractable problem of simultaneously trading off exploration and exploitation [27].

[2] Accounting for a nominal feedback controller can be used to reduce the size of this tube and is a simple extension [5, 7, 14, 15]. In this work, we omit feedback to better demonstrate the adaptation capabilities of the meta-training model, the tightness of the confidence sets, and to better verify safety claims of the framework.

[3] As we are using a sampling-based approach to compute the reachable sets [7], their variance could be used as a proxy for the variance of $l(\boldsymbol{x}, \boldsymbol{u})$, and account for the variance of the cost, or minimize its risk [26].

Algorithm 1 Sequential Exploration and Exploitation for Learning Safely (**SEELS**)

Input: Meta-training model satisfying A.3 and A.4

```
while $x_0 \notin \mathcal{X}_{\text{goal}}$ do
    for $N_i \in \{\underline{N}_{\text{reach}}, \dots, \overline{N}_{\text{reach}}\}$ do    ▷ Try reaching
        $(\mu, u) \leftarrow$ Solve (Reach-OCP)
        if (Reach-OCP) feasible then
            Apply $u_{0:N-1}$ to true system    ▷ Reach
            Break
    for $N_i \in \{1, \dots, N_{\text{info}}\}$ do    ▷ Explore
        $(\mu^i, u^i) \leftarrow$ Solve (Explore-OCP)
        if (Explore-OCP) feasible then
            Compute $l^i_{\text{info}}(\mu^i, u^i)$
    $i_{\text{best}} \leftarrow \arg\max_i l^i_{\text{info}}(\mu^i, u^i)$    ▷ Get best $N$
    Apply $u^{i_{\text{best}}}$ to true system
    Update $(k, \Lambda)$ with $\{(x_k, u_k, x_{k+1})\}_{k=0}^{N-1}$
    $x_0 \leftarrow x_N$
```

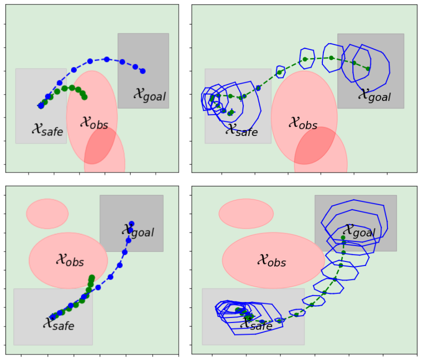

Figure 2: Rollouts on the system considered in experiments: **Left**: Due to high uncertainty, attempting to reach $\mathcal{X}_{\text{goal}}$ is initially unsafe, violating velocity and final constraints. **Right**: Using **SEELS**, the system safely reaches the goal after safely learning its dynamics.

Concretely, we write the trajectory optimization problems associated with each phase as:

$$\textbf{(Explore-OCP)} \qquad\qquad \textbf{(Reach-OCP)}$$

$$\min_{\mu, u} \sum_{k=0}^{N} l_{\text{info}}(\mu_k, u_k) \quad \text{s.t.} \quad \mathcal{X}_N^{t,\delta} \subset \mathcal{X}_0, \qquad \min_{\mu, u} \sum_{k=0}^{N} l_{\text{reach}}(\mu_k, u_k) \quad \text{s.t.} \quad \mathcal{X}_N^{t,\delta} \subset \mathcal{X}_{\text{goal}},$$

$$\bigwedge_{k=0}^{N} \begin{pmatrix} \mathcal{X}_k^{t,\delta} \subset \mathcal{X}_{\text{free}} \\ \cap \, u_k \in \mathcal{U} \end{pmatrix}, \quad \mathcal{X}_0^{t,\delta} = \{x(t)\}, \qquad \bigwedge_{k=0}^{N} \begin{pmatrix} \mathcal{X}_k^{t,\delta} \subset \mathcal{X}_{\text{free}} \\ \cap \, u_k \in \mathcal{U} \end{pmatrix}, \quad \mathcal{X}_0^{t,\delta} = \{x(t)\},$$

where $\{\mathcal{X}_k^{t,\delta}\}_{k=1}^{N}$ satisfy (7), and are computed using the confidence sets (5).

(**Reach-OCP**) uses the cost function associated with the task, and $\mathcal{X}_{\text{goal}}$ as the desired goal set. (**Explore-OCP**) is similar, but instead uses $\mathcal{X}_0$ as the goal set, thus ensuring the system will be safe for the next phase, and uses an information gathering cost l_{info} to encourage visiting states which reduce remaining uncertainty in the dynamics. In this work, we derive l_{info} from the mutual information between the unknown dynamics and the observations. The specific formula and derivation for our meta-learning model are provided in the appendix. Notably, this loss does not suffer from computational complexity that scales with the amount of data, as is the case for similar objectives derived for kernel Gaussian processes [5, 28, 29].

Our approach (**SEELS**), summarized in Algorithm 1, consists of sequentially learning a model of the dynamics by solving (**Explore-OCP**), before reaching $\mathcal{X}_{\text{goal}}$ whenever (**Reach-OCP**) admits a feasible solution.

5.2 Probabilistic Safety and Feasibility Guarantees of the Framework

This idealized algorithm enjoys probabilistically guaranteed safety and feasibility at all times, following the results from Section 4,

Theorem 2 (Probabilistic Safety)**.** *Compute confidence sets for model parameters using* (4) *and* (5)*. Using these confidence sets, compute probabilistic reachable sets* $\{\mathcal{X}_k^{t,\delta}\}_{k=1}^{N}$ *satisfying* (7)*. Using these sets, apply Algorithm 1 and sequentially solve (**Explore-OCP**) and (**Reach-OCP**).*

*Then, assuming that (**Reach-OCP**) is feasible at some time* t*, and under Assumptions 3 and 4, the system is guaranteed to satisfy* (1c)*, i.e., to be safe at all times greater than* t *and eventually reach* $\mathcal{X}_{goal}$ *with probability* $(1-\delta)$*.*

This result relies on (**Reach-OCP**) becoming feasible at some point. If the original problem is feasible with perfect knowledge of the dynamics, we can ensure this assumption is satisfied by guar-

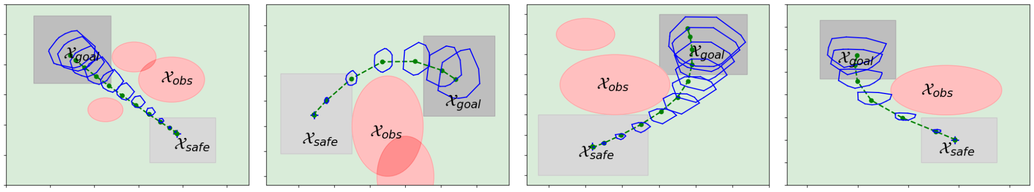

Figure 3: Initially, uncertainty is too high to safely reach the goal, Instead, we plan safe information-gathering trajectories to infer the dynamics and reduce uncertainty (see Figure 1). Once planning to $\mathcal{X}_{\text{goal}}$ is feasible, the robot can safely reach the goal while satisfying all constraints. We evaluate the safety of our framework on 4 problems (shown above) with different initial and final conditions, as well as obstacle layouts.

anteeing that the objective used for *exploration* leads to actions that continually reduce uncertainty in dynamics, related to conditions on observability and persistence of excitation [30, 31, 20].

The feasibility of (**Explore-OCP**) is guaranteed during exploration:

Theorem 3 (Probabilistic Feasibility)**.** *Under Assumptions 2, 3 and 4, there exists an optimization horizon N ensuring the feasibility of (**Explore-OCP**) at all times with probability* $(1-\delta)$.

Note that by assuming bounded noise $\boldsymbol{\epsilon}$, and exploiting confidence sets over parameters which hold jointly for all times with high probability, we can guarantee feasibility. This is in contrast to related work in the MPC literature which provides probabilistic feasibility over a finite horizon [32].

5.3 Practical Considerations

Implementation of the idealized algorithm is complicated by (1) challenges in reachability analysis and (2) challenges in nonconvex optimization.

Reachability Analysis: Computing the reachable sets in (7) is difficult due to the nonconvexity of the features $\boldsymbol{\phi}$. Methods reasoning about single step set propagation (e.g., [5]) are generally too conservative [7], as they do not account for time correlations introduced by the parameters $\boldsymbol{k}_i$. Moreover, the updates to the parameters $\boldsymbol{k}_i$ preclude exact methods which perform computations offline to characterize reachable sets, e.g., [33, 34] . Finally, methods using Lipschitz continuity to conservatively propagate these sets [5] are also too conservative in practice, and come with further limitations [7]. In this work, we leverage **randUP**, a recently derived sampling-based uncertainty propagation scheme for reachability analysis [7]. By sampling parameters $(\boldsymbol{k}_i, \boldsymbol{\epsilon}_k)$ within their confidence sets, computing reachable states $\boldsymbol{x}$ for these parameters, and approximating the reachable sets in (7) by their convex hull, it provides a scalable approach to efficiently compute these tubes with no assumptions on the system apart from differentiability. Although this method lacks finite time guarantees of safety, asymptotic guarantees can be derived using random set theory, and finite-time approximations are generally sufficient to ensure empirical safety, as demonstrated in the results.

Optimization-based planning: Using **randUP** [7], we reframe a generally intractable stochastic problem into (**Explore-OCP**) and (**Reach-OCP**), which are nonconvex optimal control problems. Efficiently computing solutions is an active field of research, which we address through a direct method based on sequential convex programming (SCP) [14]. By solving a sequence of convexified versions of the original nonconvex problem, SCP-based methods can run in real time and provide theoretical guarantees of local optimality [35, 14], given an initialization within the correct homotopy class [7]. In this work, we initialize each method with an infeasible straight-line trajectory.

Additionally, due to uncertainty, the feasibility of each problem depends on the optimization horizon N. For this reason, we perform a search over a predefined range of planning horizons. For exploitation, we select the first feasible solution if one exists, although other criteria could be used, e.g., minimal control cost. For exploration, we select the trajectory which leads to the largest expected information gain. Indeed, due to tight control constraints and safety constraints, a larger horizon does not necessarily lead to higher information gain. This heuristic works well in practice, and future work will consist of adopting a continuous time problem formulation with free final time, which is an active field of research [36].

6 Related Work

In contrast to model-free approaches to reinforcement learning, model-based methods (generally) provide better sample efficiency while enabling guarantees on constraint satisfaction and stability [37, 3]. These model-based methods rely on the choice of dynamics model parameterization—for example, neural networks [1], Gaussian processes (GPs) [3], or linear models [31]—each with associated strengths and weaknesses. Recent work in the controls community has leveraged behavioral systems theory to guarantee stability and probabilistic constraints satisfaction of a non-parametric MPC scheme [38, 30, 39]. Although such methods have been shown to perform well for nonlinear systems [31], their guarantees currently do not extend beyond time invariant linear systems. Moreover, these approaches rely on linear models, limiting their expressiveness and potentially reducing their applicability in diverse scenarios as well as generalization across scenarios.

Nonlinear controllers leveraging a neural network model of the system can provide stability guarantees [40], under smoothness and other assumptions. However, these methods require collecting a dataset for a single system (i.e., already being able to solve the task), and would need total retraining if the environment or the system change. Training a neural network dynamics model from scratch for each environment is prohibitively expensive in terms of data requirements. Our approach combines neural network features with linear online adaptation to obtain the best of both models: the linear learning is sample efficient and enables strong guarantees on performance, while the neural network features are highly expressive and enable generalization across environments. While prior work has leveraged meta-learning for fast online adaptation [41], such approaches are difficult to provide safety guarantees for, as they typically adapt using online gradient descent in non-convex problems. In contrast, our linear online adaptation enables construction of confidence sets for model parameters that hold throughout the learning process.

Gaussian processes have been widely used for safe learning-based control and exploration, as they can represent any nonlinear function in a bounded reproducing kernel Hilbert space (RKHS). GPs are nonlinear, Bayesian models that obtain sample efficiency through exact conditioning and reasonably expressive features through the choice of kernel [28]. While bounds providing similar guarantees to Theorem 1 can be derived for GPs, such bounds are generally too conservative, in which case the authors usually set these constants to arbitrary values in experiments [25, 5]. Alternatively, assuming that the RKHS is known, and that any function in this space lies in the span of finite-dimensional features ϕ is common in practice [20, 21]. Importantly, this linear structure enables the derivation of bounds over models [42], which we use directly in this work. In contrast with prior work, we explicitly learn features and quantify prior uncertainty in an offline meta-training procedure [6, 8], enabling us to design a model which is calibrated and accurate enough to represent possible systems, and allows verifying that representation error is small *offline*, before deploying this system.

7 Numerical Results: Safely Transporting an Uncertain Payload

We verify our proposed approach on a nonlinear six-dimensional planar free-flyer robot navigating in a cluttered environment. The goal consists of safely transporting an uncertain payload, which causes a change in mass and inertial properties (including the location of the center of mass), to a goal region. The robot has three control inputs with limited authority: two pairs of gas thrusters, and a reaction wheel. The complete problem formulation is provided in the appendix.

We set obstacle avoidance constraints $\boldsymbol{x}_k \notin \mathcal{X}_{\text{obs}}$, which we reformulate using the signed distance function [14]. We directly use our theoretically computed bounds within our safe learning algorithm, i.e., we use (4) to sample model parameters, and the bound on ϵ. To validate the safety and reliability of our framework, we run it on a batch of 250 problems with different parameters realizations of the dynamics. We also randomize scenarios with 4 different obstacle configurations, and initial and final conditions, and compare the sensitivity to the noise magnitude, to the number of samples for reachability analysis, to δ, and to the regularization of β_i. Figure 3 shows illustrative experiments, and we release all data in the supplementary material, summarized in Tables 1, 2 and 3.

Discussion: Results in Table 1 show that our framework can reliably solve this problem for multiple obstacle fields, while guaranteeing probabilistic safety. In particular, the joint chance constraint (1c) is conservatively satisfied in practice, and the system reaches the goal safely after a few exploration phases. We compare with a method which only considers uncertainty from the additive disturbances

	δ	# Explore	$\boldsymbol{x} \notin \mathcal{X}_{\text{obs}}$	$\boldsymbol{x} \in \mathcal{X}_{\text{min/max}}$	$\boldsymbol{x}_N \in \mathcal{X}_{\text{goal}}$	$\boldsymbol{x} \in \mathcal{X}_{\text{all}}$
SEELS	0.1	2.3 ± 0.01	$97.6 \pm 1.9\%$	$97.6 \pm 1.9\%$	$96.8 \pm 2.2\%$	$93.2 \pm 3.1\%$
Mean-Equivalent	-	0	$39.6 \pm 6.0\%$	$99.6 \pm 0.8\%$	$22.8 \pm 5.2\%$	$19.6 \pm 4.9\%$

Table 1: Results for 250 randomized experiments. For each experiment, we report the number of exploration phases, check constraints satisfaction, and report the percentage of experiments for which all constraints are satisfied ($\boldsymbol{x} \in \mathcal{X}_{\text{all}}$), with 95% confidence intervals. We run a mean-equivalent version of **SEELS** (Algorithm 1) which accounts for the disturbances $\boldsymbol{\epsilon}_k$, but does not consider model uncertainty. Our framework is guaranteed to simultaneously respect all constraints $(1 - \delta)$ fraction of the time.

$\boldsymbol{\epsilon}_k$, in which case the system deems reaching $\mathcal{X}_{\text{goal}}$ directly to be safe. This naive approach violates safety constraints in most cases, which demonstrates the need for sequential online learning to reliably solve this problem.

Sensitivity to parameters: We perform further experiments for high values of $\sigma_{\boldsymbol{\epsilon}}$, summarized in Figure 4. First, we observe that increasing M does lead to increased success rate and probability of safety. Therefore, by Theorems 2 and 3, success is guaranteed as long as the number of samples for reachability analysis M is high enough. We refer to [7] for further discussion, evaluation, and possible extensions. Second, the conservatism of the algorithm can be tuned by choosing a different value for δ. In particular, by opting for lower probability of safety, $\mathcal{X}_{\text{goal}}$ is reached faster in average. Finally, we observe that regularizing β_i reduces conservatism, while still guaranteeing probabilistic safety in practice. This correlates both with less conservatism, and faster time to reach $\mathcal{X}_{\text{goal}}$.

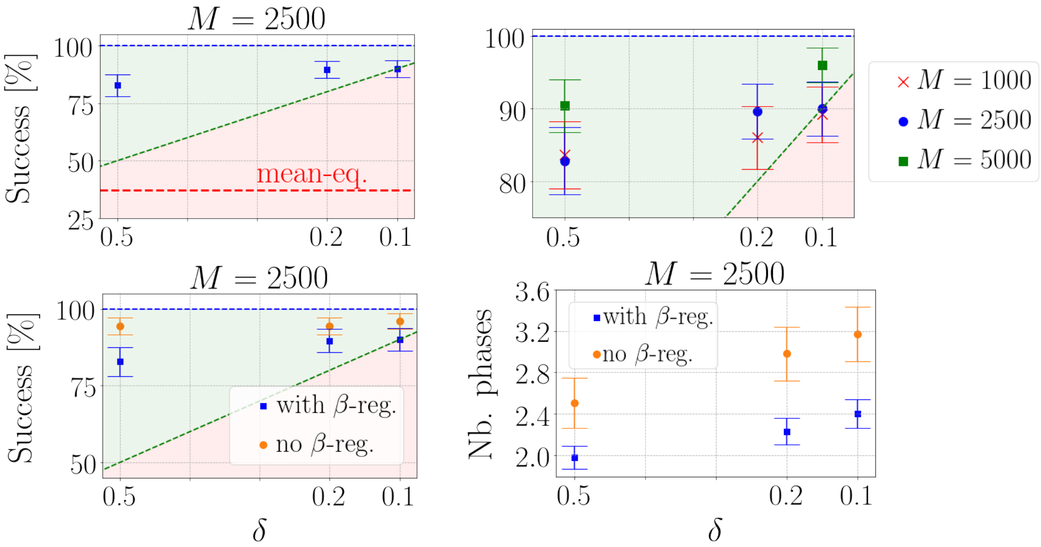

Figure 4: Results for 250 randomized experiments, different parameters, and high noise levels $\boldsymbol{\epsilon}_k$. On plots showing success percentages (all constraints are satisfied and $\boldsymbol{x}_N \in \mathcal{X}_{\text{goal}}$), the green region denotes results where the success percentage is at or above the desired probability of success given by δ, and the red region indicates where the true probability of success is lower than desired. Error bars correspond to 95% confidence intervals.

8 Conclusion

To safely perform tasks under high initial uncertainty, we presented **SEELS**: a framework which sequentially explores to learn the properties of the system, while guaranteeing safety at all times through a single joint chance constraint. The key consists of leveraging Bayesian meta-learning to encode prior information about the system, and to ensure efficient and interpretable online learning. This enables the definition of confidence sets over the learned parameters, which are then used for uncertainty-aware planning to guarantee feasibility and satisfaction of all constraints at all times with high probability. We demonstrated the reliability of our approach through extensive simulations, by randomizing the parameters of the system, boundary conditions, and obstacles, studied the sensitivity to different hyper-parameters, and proposed methods to improve performance and reduce conservatism.

Future work: We plan on verifying our framework on hardware experiments, with a focus on satisfying Assumptions 3 and 4. Further, we will explore methods to compute invariant sets satisfying 2 by leveraging the prior of the meta-learning model. Extensions of [7] will strengthen the safety guarantees to the case of a finite number of samples for uncertainty propagation, and work on free final time trajectory optimization will replace the search over horizons N in our algorithm. Finally, we plan on investigating regret bounds for such constrained problems, to provide guarantees on the time required to perform a task given known geometric properties of the problem [43].

Acknowledgments

We would like to thank Joseph Lorenzetti for his helpful feedback and suggestions.

References

[1] S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17:1–40, 2016.

[2] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26), 2019.

[3] M. Deisenroth, D. Fox, and C. Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 37(2): 408–423, 2015.

[4] S. Dean, S. Tu, N. Matni, and B. Recht. Safely learning to control the constrained linear quadratic regulator. In *American Control Conference*, 2019.

[5] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause. Learning-based model predictive control for safe exploration. In *Proc. IEEE Conf. on Decision and Control*, 2018.

[6] J. Harrison, A. Sharma, and M. Pavone. Meta-learning priors for efficient online bayesian regression. In *Workshop on Algorithmic Foundations of Robotics*, 2018.

[7] T. Lew and M. Pavone. Sampling-based reachability analysis: A random set theory approach with adversarial sampling, 2020. Available at `https://arxiv.org/abs/2008.10180`.

[8] J. Harrison, A. Sharma, R. Dyro, X. Wang, R. Calandra, and M. Pavone. Control adaptation via meta-learning dynamics. In *NeurIPS Workshop on Meta-Learning*, 2018.

[9] J. Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.

[10] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Int. Conf. on Machine Learning*, 2017.

[11] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap. Meta-learning with memory-augmented neural networks. In *Int. Conf. on Machine Learning*, 2016.

[12] S. Hochreiter, A. S. Younger, and P. R. Conwell. Learning to learn using gradient descent. In *International Conference on Artificial Neural Networks*, 2001.

[13] L. Hewing, A. Carron, K. P. Wabersich, and M. N. Zeilinger. On a correspondence between probabilistic and robust invariant sets for linear systems. In *European Control Conference*, 2018.

[14] T. Lew, R. Bonalli, and M. Pavone. Chance-constrained sequential convex programming for robust trajectory optimization. In *European Control Conference*, 2020.

[15] L. Hewing, J. Kabzan, and M. N. Zeilinger. Cautious Model Predictive Control using Gaussian Process Regression. *IEEE Transactions on Control Systems Technology*, 2018. Early Access.

[16] K. Polymenakos, L. Laurenti, A. Patane, J. P. Calliess, L. Cardelli, M. Kwiatkowska, A. Abate, and S. Roberts. Safety guarantees for planning based on iterative Gaussian processes, 2020. Available at `https://arxiv.org/abs/1912.00071`.

[17] M. J. Khojasteh, V. Dhiman, M. Franceschetti, and N. Atanasov. Probabilistic safety constraints for learned high relative degree system dynamics. In *2nd Annual Conference on Learning for Dynamics & Control*, 2020.

[18] R. Cheng, M. J. Khojasteh, A. D. Ames, and J. W. Burdick. Safe multi-agent interaction through robust control barrier functions with learned uncertainties, 2020. Available at `https://arxiv.org/abs/2004.05273`.

[19] E. Schmerling and M. Pavone. Evaluating trajectory collision probability through adaptive importance sampling for safe motion planning. In *Robotics: Science and Systems*, 2017.

[20] H. Mania, M. I. Jordan, and B. Recht. Active learning for nonlinear system identification with guarantees, 2020. Available at https://arxiv.org/abs/2006.10277.

[21] S. Kakade, A. Krishnamurthy, K. Lowrey, M. Ohnishi, and W. Sun. Information theoretic regret bounds for online nonlinear control, 2020. Available at https://arxiv.org/abs/2006.12466.

[22] R. Amit and R. Meir. Meta-learning by adjusting priors based on extended pac-bayes theory. In *Int. Conf. on Machine Learning*, 2018.

[23] V. Kuleshov, N. Fenner, and S. Ermon. Accurate uncertainties for deep learning using calibrated regression. In *Int. Conf. on Machine Learning*, 2018.

[24] F. Berkenkamp. *Safe Exploration in Reinforcement Learning: Theory and Applications in Robotics*. PhD thesis, Institute for Machine Learning, ETH Zürich, 2018.

[25] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause. Safe model-based reinforcement learning with stability guarantees. In *Conf. on Neural Information Processing Systems*, 2017.

[26] S. Singh, Y.-L. Chow, A. Majumdar, and M. Pavone. A framework for time-consistent, risk-sensitive model predictive control: Theory and algorithms. *IEEE Transactions on Automatic Control*, 64(7):2905–2912, 2018.

[27] Y. Bar-Shalom and E. Tse. Dual effect, certainty equivalence, and separation in stochastic control. *IEEE Transactions on Automatic Control*, 19(5), 1974.

[28] C. K. Williams and C. E. Rasmussen. *Gaussian processes for machine learning*. MIT press, 2006.

[29] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Int. Conf. on Machine Learning*, 2010.

[30] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer. Robust constraint satisfaction in data-driven MPC, 2020. Available at https://arxiv.org/abs/2003.06808.

[31] J. Coulson, J. Lygeros, and F. Dörfler. Data-enabled predictive control: In the shallows of the DeePC. 2018.

[32] M. Ono. Joint chance-constrained model predictive control with probabilistic resolvability. In *American Control Conference*, 2012.

[33] S. Bansal, S. L. Chen, M. Herbert, and C. J. Tomlin. Hamilton-Jacobi reachability: A brief overview and recent advances. In *Proc. IEEE Conf. on Decision and Control*, 2017.

[34] D. D. Fan, A. Agha-mohammadi, and E. A. Theodorou. Deep learning tubes for tube MPC. In *Robotics: Science and Systems*, 2020.

[35] Y. Mao, M. Szmuk, and B. Açikmeşe. Successive convexification of non-convex optimal control problems and its convergence properties. In *Proc. IEEE Conf. on Decision and Control*, 2016.

[36] R. Bonalli, A. Cauligi, A. Bylard, and M. Pavone. GuSTO: guaranteed sequential trajectory optimization via sequential convex programming. In *Proc. IEEE Conf. on Robotics and Automation*, 2019.

[37] B. Recht. A tour of reinforcement learning: The view from continuous control. *Annual Review of Control, Robotics, and Autonomous Systems*, 2(1):253–279, 2019.

[38] J. Coulson, J. Lygeros, and F. Dörfler. Distributionally robust chance constrained data-enabled predictive control, 2020. Available at https://arxiv.org/abs/2006.01702.

[39] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer. Data-driven model predictive control with stability and robustness guarantees, 2020. Available at https://arxiv.org/abs/1906.04679.

[40] G. Shi, X. Shi, M. O'Connell, R. Yu, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S.-J. Chung. Neural lander: Stable drone landing control using learned dynamics. In *Proc. IEEE Conf. on Robotics and Automation*, 2019.

[41] A. Nagabandi, I. Clavera, L. Simin, R. S. Fearing, P. Abbeel, S. Levine, and C. Chelsea Finn. Learning to adapt in dynamic real-world environments through meta-reinforcement learning. In *Int. Conf. on Learning Representations*, 2019.

[42] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári. Improved algorithms for linear stochastic bandits. In *Conf. on Neural Information Processing Systems*, 2011.

[43] M. Kleinbort, K. Solovey, Z. Littlefield, K. E. Bekris, and D. Halperin. Probabilistic completeness of RRT for geometric and kinodynamic planning with forward propagation. *IEEE*

Robotics and Automation Letters, 4(2):10–16, 2018.

[44] D. J. C. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992.

[45] A. Chowdhury, S. R. Gopalan. On kernelized multi-armed bandits. In *Int. Conf. on Machine Learning*, 2017.

[46] L. Fluckiger, K. Browne, B. Coltin, J. Fusco, T. Morse, and A. Symington. Astrobee robot software: Enabling mobile autonomy on the iss. In *Int. Symp. on Artificial Intelligence, Robotics and Automation in Space*, 2018.

[47] M. Ekal and R. Ventura. On the accuracy of inertial parameter estimation of a free-flying robot while grasping an object. *Journal of Intelligent & Robotic Systems*, pages 1–11, 2019.

A Further Algorithmic Details

A.1 Regularizing Meta-training for Safe Online Learning

The size of the confidence sets for the model parameters $\boldsymbol{k}_i$ is controlled by the term β_i, which depends on the structure of the problem. Specifically, by relying on the expressiveness of the meta-learned features $\boldsymbol{\phi}(\cdot,\cdot)$, parameterized by a feed-forward neural network, different set of weights for $\boldsymbol{\phi}$ and prior parameters $(\bar{\boldsymbol{k}}_{i,0}, \boldsymbol{\Lambda}_{i,0})$ could be used to parameterize the unknown dynamics, while satisfying Assumptions 3 and 4. Therefore, it is possible to modify the meta-training procedure to obtain a model with lower values of β_i, and improve performance without compromising safety.

Specifically, we note from (4) that the value of β_i depends on the ratio between the maximum and minimum eigenvalues of the prior and posterior precision matrices $\boldsymbol{\Lambda}_i$. If $\lambda_{\max}(\boldsymbol{\Lambda}_{i,0}) \leq 1$ as is typically the case in our experiments, then it holds that

$$\begin{aligned}\lambda_{\max}(\boldsymbol{\Lambda}_{i,0})/\lambda_{\min}(\boldsymbol{\Lambda}_{i,t}) = \lambda_{\max}(\boldsymbol{\Lambda}_{i,t}^{-1})/\lambda_{\min}(\boldsymbol{\Lambda}_{i,0}^{-1}) &\leq \lambda_{\max}(\boldsymbol{\Lambda}_{i,t}^{-1})\lambda_{\min}(\boldsymbol{\Lambda}_{i,0}^{-1}) \\ &\leq \lambda_{\max}(\boldsymbol{\Lambda}_{i,t}^{-1})\lambda_{\max}(\boldsymbol{\Lambda}_{i,0}^{-1}).\end{aligned}$$

Furthermore, $\lambda_{\max}(\boldsymbol{\Lambda}) \leq \sqrt{\mathrm{Tr}(\boldsymbol{\Lambda}^T\boldsymbol{\Lambda})}$. Combining with the above, we propose to regularize an upper bound of the ratio $\lambda_{\max}(\boldsymbol{\Lambda}_{i,0})/\lambda_{\min}(\boldsymbol{\Lambda}_{i,t})$ during offline meta-training:

$$\mathcal{L}_{\text{reg}}(\boldsymbol{\Lambda}_{i,0}) = \alpha_{\text{reg}} \sum_{i=1}^{n} \mathrm{Tr}(\boldsymbol{\Lambda}_{i,t}^{-T}\boldsymbol{\Lambda}_{i,t}^{-1})\mathrm{Tr}(\boldsymbol{\Lambda}_{i,0}^{-T}\boldsymbol{\Lambda}_{i,0}^{-1}) \tag{11}$$

where the scalar α_{reg} controls the strength of this regularization, and is selected using a validation dataset. As the meta-training model is directly parameterized by the inverse of the precision matrices $\boldsymbol{\Lambda}_i$ [8], this regularizer can easily be added to the standard training loss.

From (4), we observe that β_i also depends on the ratio of determinants of the prior and posterior precision matrices $\big(\det(\boldsymbol{\Lambda}_{i,t})/\det(\boldsymbol{\Lambda}_{i,0})\big)$. Although a convex regularizer for this term can be derived, we found that including it did not lead to performance improvements. This ratio can be interpreted as capturing the amount of information that the model has gathered online, which is independent of the structure of the prior model. Before learning, this ratio is 1, so the other term composed of the ratio of eigenvalues dominates β_i. We observed that it is during these early stages that the meta-training model and its bounds β_i are most conservative, which could explain the importance of the regularizer in (11), whereas regularizing the ratio of determinants appears to make little difference.

A.2 Information cost

During the exploration phase, we perform trajectory optimization with an objective function that encourages visiting states and taking actions that reduce uncertainty over the unknown dynamics. To do so, a natural objective function to maximize is the mutual information between the unknown function $\boldsymbol{g}(\cdot,\cdot,\boldsymbol{\theta})$ and the observations $\tilde{\boldsymbol{x}}^+ = \boldsymbol{x}_t - \boldsymbol{h}(\boldsymbol{x}_{t-1}, \boldsymbol{u}_{t-1})$. This cost characterizes the *information gain* [44, 29, 45] from observing $\tilde{\boldsymbol{x}}^+$.

We derive this objective for the linear-Gaussian Bayesian model assumed by the meta-learning formulation in [6]. For this formulation, which assumes that observations are corrupted with Gaussian noise, the mutual information can be computed in closed form. While in this work we assume bounded (non-Gaussian) noise corrupting our measurements, we find that making this approximation works well in practice to encourage exploration.

Let the posterior distribution over models be specified by $\boldsymbol{k}_i \sim \mathcal{N}(\bar{\boldsymbol{k}}_i, \sigma_{\epsilon_i}^2 \boldsymbol{\Lambda}_i)$, with Gaussian-distributed observation noise ϵ_i of variance $\sigma_{\epsilon_i}^2$. In this setting, the marginal distribution over observations $\boldsymbol{x}_i^+ = \boldsymbol{k}_i \boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{u}) + \epsilon_i$ given an arbitrary state $\boldsymbol{x}$ and control input $\boldsymbol{u}$ is also normally distributed as $\mathcal{N}(\boldsymbol{k}_i \boldsymbol{\phi}, (1 + \boldsymbol{\phi}^T \boldsymbol{\Lambda}_i^{-1} \boldsymbol{\phi}) \sigma_{\epsilon_i}^2)$, where $\boldsymbol{\phi} = \boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{u})$.

Next, we define the mutual information $\mathcal{I}$ between the observation $\boldsymbol{x}^+$, and the true model $\boldsymbol{g}(\cdot, \cdot, \boldsymbol{\theta})$, as a function of the current state $\boldsymbol{x}$ and control input $\boldsymbol{u}$, and assuming that Assumption 3 holds. This quantity denotes the information gain from applying the control input $\boldsymbol{u}$ to the true system from $\boldsymbol{x}$, and observing $\boldsymbol{x}^+$ to update our model. The mutual information is defined using the entropy $\mathcal{H}(\cdot)$, which for a Gaussian-distributed random variable $\boldsymbol{x}^+ \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ evaluates to $\mathcal{H}(\boldsymbol{x}) = (1/2)\log(\det(2\pi e \boldsymbol{\Sigma}))$. Hence, the information gain from observing the scalar random variable $\boldsymbol{x}_i^+$ can be expressed as:

$$\begin{aligned} \mathcal{I}(\boldsymbol{x}_i^+; \boldsymbol{g}_{\boldsymbol{\theta}}) = \mathcal{H}(\boldsymbol{x}_i^+) - \mathcal{H}(\boldsymbol{x}_i^+ | \boldsymbol{g}_{\boldsymbol{\theta}}) &= \frac{1}{2}\Big(\log(\mathrm{var}(\boldsymbol{x}_i^+)) - \log(\mathrm{var}(\boldsymbol{x}_i^+ | \boldsymbol{g}_{\boldsymbol{\theta}})\Big) \\ &= \frac{1}{2}\Big(\log((1 + \boldsymbol{\phi}^T \boldsymbol{\Lambda}_i^{-1} \boldsymbol{\phi}) \sigma_{\epsilon_i}^2)) - \log(\sigma_{\epsilon_i}^2))\Big) = \frac{1}{2}\Big(\log(1 + \boldsymbol{\phi}^T \boldsymbol{\Lambda}_i^{-1} \boldsymbol{\phi})\Big). \end{aligned}$$

For our problem formulation, this quantity approximately expresses the information gain from observing each dimension i of the state (which are modeled independently in our formulation, see (2)). Intuitively, we would like to design exploration trajectories that visit states and take actions where this quantity is high for all dimensions of the state, as these observations would be the most informative in terms of reducing uncertainty over the underlying model. Thus, we use this term to guide the exploration phases, and optimize for the objective

$$l_{\text{info}}(\boldsymbol{x}, \boldsymbol{u}; \boldsymbol{\Lambda}_{1,t}, \dots, \boldsymbol{\Lambda}_{n,t}) = \frac{1}{2} \sum_{i=1}^{n} \log(1 + \boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{u})^T \boldsymbol{\Lambda}_{i,t}^{-1} \boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{u})). \tag{12}$$

Note that this is a function of the current information state of the model, specified by the updated precision matrices $\boldsymbol{\Lambda}_{1,t}, \dots, \boldsymbol{\Lambda}_{n,t}$. This provides an objective which encourages exploring states in the feature space spanned by $\boldsymbol{\phi}(\cdot, \cdot)$ which have highest variance, to quickly reduce uncertainty.

Note that the expected information gain along a trajectory is not simply the sum of the expected information gains per transition, as expressed in (**Explore-OCP**) when summing (12) over $k = 0, \dots, N$. However, correctly computing the expected information gain along the trajectory would require factoring in model updates along the trajectory; we find that considering the sum of single-transition information gain with the current precision matrices $\boldsymbol{\Lambda}_{i,t}$ is sufficient in guiding exploration for our work. The problem of optimal exploration is beyond the scope of this framework.

B Proofs

B.1 Proof of Theorem 1: Uniformly Calibrated Confidence Sets

The proof of Theorem 1 follows from the proof of [42, Theorem 2], by making substitutions accordingly for our meta-learning model. To do so, we use the following lemma, which follows from [42, Theorem 1], by considering each dimension $i = 1, \dots, n$ of the meta-learning model independently.

Lemma 1 (Self-Normalized Bound for Vector-Valued Martingales)**.** *Let $\{\mathcal{F}_t\}_{t=0}^{\infty}$ be a filtration. Define $\{\epsilon_t^i\}_{t=1}^{\infty}$, a real-valued stochastic process such that ϵ_t^i is $\mathcal{F}_t$-measurable, and conditionally σ_{ϵ_i}-subgaussian. Let $\{\boldsymbol{\phi}_t\}_{t=1}^{\infty}$ be a $\mathbb{R}^d$-valued stochastic process such that $\boldsymbol{\phi}_t$ is $\mathcal{F}_{t-1}$-measurable.*

Let $\boldsymbol{\Lambda}_{i,0}$ be a $d \times d$ positive definite matrix, and define $\boldsymbol{\Lambda}_{i,t}$ as in (3). Further, for any $t \geq 0$, define $\mathbf{S}_t = \sum_{s=1}^{t} \epsilon_s^i \boldsymbol{\phi}_s$. Then, for any $\delta_i > 0$, with probability at least $(1 - \delta_i)$, for all $t \geq 0$,

$$\|\mathbf{S}_t\|_{\boldsymbol{\Lambda}_{i,t}^{-1}}^2 \leq 2\sigma_{\epsilon_i}^2 \log\left(\frac{1}{\delta} \frac{\det(\boldsymbol{\Lambda}_{i,t})^{1/2}}{\det(\boldsymbol{\Lambda}_{i,0})^{1/2}} \right) \tag{13}$$

Proof. The filtration $\{\mathcal{F}_t\}_{t=0}^{\infty}$ is defined by considering the σ-algebra $\mathcal{F}_t = \sigma(\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_{t+1}, \boldsymbol{\epsilon}_0, \dots, \boldsymbol{\epsilon}_t)$, where $\boldsymbol{\phi}_t = \boldsymbol{\phi}(\boldsymbol{x}_t, \boldsymbol{u}_t)$, and the $\boldsymbol{x}_t$ are given by (1b). Then, this result follows by direct application of [42, Theorem 1], substituting $(X, \eta, \theta, \bar{V}_t, V)$ with $(\boldsymbol{\phi}, \epsilon^i, \boldsymbol{k}_i, \boldsymbol{\Lambda}_{i,t}, \boldsymbol{\Lambda}_{i,0})$. □

We stress that this result holds jointly for all times $t \geq 0$, such that $\mathbb{P}((13)) \geq (1-\delta_i)$. This result is key to ensure joint chance constraint satisfaction, and guarantee safety and feasibility of our framework.

Next, we prove Theorem 1, which we restate here for completeness.

Theorem 1 (Uniformly Calibrated Confidence Sets)**.** *Consider the true system* (1b)*,*

$$\boldsymbol{x}_{k+1} = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}) + \boldsymbol{\epsilon}_k,$$

where $\boldsymbol{\epsilon}_k$ *is* σ_ϵ*-subgaussian and bounded. Consider the meta-learning model* (2)*, given as* $\hat{\boldsymbol{g}}(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{K}\boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{u})$*, where* $\boldsymbol{\phi} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^d$*, and* $\boldsymbol{K}$ *is an* $n \times d$ *matrix, with* n *rows* $\boldsymbol{k}_i$*. Starting from* $(\bar{\boldsymbol{k}}_{i,0}, \boldsymbol{\Lambda}_{i,0})$*, with* $\bar{\boldsymbol{k}}_{i,0} \in \mathbb{R}^d$*, and* $\boldsymbol{\Lambda}_{i,0}$ *a* $d \times d$ *positive definite matrix, define the sequence* $\{(\bar{\boldsymbol{k}}_{i,s}, \boldsymbol{\Lambda}_{i,s})\}_{s=0}^t$*, where* $(\bar{\boldsymbol{k}}_{i,t}, \boldsymbol{\Lambda}_{i,t})$ *is computed with online data from* (1b) *using* (3) *as*

$$\boldsymbol{\Lambda}_{i,t} = \Phi_{t-1}^T \Phi_{t-1} + \boldsymbol{\Lambda}_{i,0}, \qquad \bar{\boldsymbol{k}}_{i,t} = \boldsymbol{\Lambda}_{i,t}^{-1}(\Phi_{t-1}^T \boldsymbol{G}_{i,t} + \boldsymbol{\Lambda}_{i,0}\bar{\boldsymbol{k}}_{i,0}), \qquad i = 1, \ldots, n,$$

$\boldsymbol{G}_t^T = [\boldsymbol{x}_1 - \boldsymbol{h}(\boldsymbol{x}_0, \boldsymbol{u}_0), \ldots, \boldsymbol{x}_t - \boldsymbol{h}(\boldsymbol{x}_{t-1}, \boldsymbol{u}_{t-1})] \in \mathbb{R}^{n \times t}$*, and* $\Phi_{t-1}^T = [\boldsymbol{\phi}(\boldsymbol{x}_0, \boldsymbol{u}_0), \ldots, \boldsymbol{\phi}(\boldsymbol{x}_{t-1}, \boldsymbol{u}_{t-1})] \in \mathbb{R}^{d \times t}$*. Further, define* $\delta_i = \delta/(2n)$*, and*

$$\beta_i(\boldsymbol{\Lambda}_{i,t}, \delta_i) = \sigma_{\epsilon_i}\left(\sqrt{2\log\left(\frac{1}{\delta_i}\frac{\det(\boldsymbol{\Lambda}_{i,t})^{1/2}}{\det(\boldsymbol{\Lambda}_{i,0})^{1/2}}\right)} + \sqrt{\frac{\lambda_{\max}(\boldsymbol{\Lambda}_{i,0})}{\lambda_{\min}(\boldsymbol{\Lambda}_{i,t})}\chi_d^2(1-\delta_i)}\right).$$

Then, under Assumptions 3 and 4,

$$\mathbb{P}\left(\|\boldsymbol{k}_i^* - \bar{\boldsymbol{k}}_{i,t}\|_{\boldsymbol{\Lambda}_{i,t}} \leq \beta_i(\boldsymbol{\Lambda}_{i,t}, \delta_i) \quad \forall t \geq 0\right) \geq (1 - 2\delta_i).$$

Proof. This proof is a straightforward extension of [42, Theorem 2], where we use Assumption 4 to provide a probabilistic error bound for the model missmatch over the prior for $\boldsymbol{k}_i^*$, Lemma 1 to bound the estimation error due to $\boldsymbol{\epsilon}_k$, and Boole's inequality to obtain the bound β_i.

Define $\boldsymbol{\epsilon}^i = (\epsilon_1^i, \ldots, \epsilon_t^i)^T$. For any $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$, and $\mathbf{A}$ a $d \times d$ positive definite matrix, define the weighted norm $\|\mathbf{a}\|_{\mathbf{A}}^2 = \mathbf{a}^T\mathbf{A}\mathbf{a}$, and weighted inner product $\langle \mathbf{a}, \mathbf{b} \rangle_{\mathbf{A}} = \mathbf{a}^T\mathbf{A}\mathbf{b}$. For conciseness, we drop the indices i and t, and denote $(\boldsymbol{k}^*, \bar{\boldsymbol{k}}, \boldsymbol{\Lambda}, \bar{\boldsymbol{k}}_0, \boldsymbol{\Lambda}_0, \Phi, \boldsymbol{\epsilon}) = (\boldsymbol{k}_i^*, \bar{\boldsymbol{k}}_{i,t}, \boldsymbol{\Lambda}_{i,t}, \bar{\boldsymbol{k}}_{i,0}, \boldsymbol{\Lambda}_{i,0}, \Phi_{t-1}, \boldsymbol{\epsilon}^i)$.

Under Assumption 3, we can write $\boldsymbol{G}_{i,t} = \Phi\boldsymbol{k}^* + \boldsymbol{\epsilon}$. Then, we rewrite the mean estimate $\bar{\boldsymbol{k}}$ of $\boldsymbol{k}^*$ at time t, as

$$\begin{aligned}
\bar{\boldsymbol{k}} &= (\boldsymbol{\Lambda}_0 + \Phi^T\Phi)^{-1}(\boldsymbol{\Lambda}_0\bar{\boldsymbol{k}}_0 + \Phi^T(\Phi\boldsymbol{k}^* + \boldsymbol{\epsilon})) \\
&= (\boldsymbol{\Lambda}_0 + \Phi^T\Phi)^{-1}\Phi^T\boldsymbol{\epsilon} + (\boldsymbol{\Lambda}_0 + \Phi^T\Phi)^{-1}(\boldsymbol{\Lambda}_0 + \Phi^T\Phi)\boldsymbol{k}^* - (\boldsymbol{\Lambda}_0 + \Phi^T\Phi)^{-1}\boldsymbol{\Lambda}_0(\boldsymbol{k}^* - \bar{\boldsymbol{k}}_0) \\
&= \boldsymbol{\Lambda}^{-1}\Phi^T\epsilon + \boldsymbol{k}^* - \boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_0(\boldsymbol{k}^* - \bar{\boldsymbol{k}}_0),
\end{aligned}$$

from which we obtain, for any $\mathbf{a} \in \mathbb{R}^d$,

$$\mathbf{a}^T(\bar{\boldsymbol{k}} - \boldsymbol{k}^*) = \langle \mathbf{a}, \Phi^T\boldsymbol{\epsilon} \rangle_{\boldsymbol{\Lambda}^{-1}} - \langle \mathbf{a}, \boldsymbol{\Lambda}_0(\boldsymbol{k}^* - \bar{\boldsymbol{k}}_0) \rangle_{\boldsymbol{\Lambda}^{-1}}. \tag{14}$$

Note that $\boldsymbol{\Lambda}_0 \succ 0$, so $\boldsymbol{\Lambda} \succ 0$, and these inner products are well defined.

By the Cauchy-Schwarz inequality,

$$\begin{aligned}
|\mathbf{a}^T(\bar{\boldsymbol{k}} - \boldsymbol{k}^*)| &\leq \|\mathbf{a}\|_{\boldsymbol{\Lambda}_t^{-1}}\left(\left\|\Phi^T\epsilon\right\|_{\boldsymbol{\Lambda}^{-1}} + \left\|\boldsymbol{\Lambda}_0(\boldsymbol{k}^* - \bar{\boldsymbol{k}}_0)\right\|_{\boldsymbol{\Lambda}^{-1}}\right) \\
&\leq \|\mathbf{a}\|_{\boldsymbol{\Lambda}^{-1}}\left(\left\|\Phi^T\epsilon\right\|_{\boldsymbol{\Lambda}^{-1}} + \sqrt{\frac{\lambda_{\max}(\boldsymbol{\Lambda}_0)}{\lambda_{\min}(\boldsymbol{\Lambda})}}\left\|\boldsymbol{k}^* - \bar{\boldsymbol{k}}_0\right\|_{\boldsymbol{\Lambda}_0}\right),
\end{aligned} \tag{15}$$

where the second inequality is obtained as

$$\begin{aligned}
\left\|\boldsymbol{\Lambda}_0(\boldsymbol{k}^* - \bar{\boldsymbol{k}}_0)\right\|_{\boldsymbol{\Lambda}^{-1}}^2 \leq \frac{\lambda_{\max}(\boldsymbol{\Lambda}^{-1})}{\lambda_{\min}(\boldsymbol{\Lambda}_0^{-1})}\left\|\boldsymbol{\Lambda}_0(\boldsymbol{k}^* - \bar{\boldsymbol{k}}_0)\right\|_{\boldsymbol{\Lambda}_0^{-1}}^2 &= \frac{\lambda_{\max}(\boldsymbol{\Lambda}_0)}{\lambda_{\min}(\boldsymbol{\Lambda})}\left\|\boldsymbol{\Lambda}_0(\boldsymbol{k}^* - \bar{\boldsymbol{k}}_0)\right\|_{\boldsymbol{\Lambda}_0^{-1}}^2 \\
&= \frac{\lambda_{\max}(\boldsymbol{\Lambda}_0)}{\lambda_{\min}(\boldsymbol{\Lambda})}\left\|\boldsymbol{k}^* - \bar{\boldsymbol{k}}_0\right\|_{\boldsymbol{\Lambda}_0}^2.
\end{aligned}$$

By Lemma 1, for any $\delta_i \geq 0$, with probability at least $(1-\delta_i)$, we have

$$\left\|\Phi^T \epsilon\right\|^2_{\mathbf{\Lambda}^{-1}} \leq 2\sigma^2_{\epsilon_i} \log\left(\frac{1}{\delta_i}\frac{\det(\mathbf{\Lambda})^{1/2}}{\det(\mathbf{\Lambda}_0)^{1/2}}\right) \quad \forall t \geq 0. \tag{16}$$

By Assumption 4, for $\delta_i = \delta/(2n)$, with probability at least $(1-\delta_i)$,

$$\|\boldsymbol{k}^* - \bar{\boldsymbol{k}}_0\|^2_{\mathbf{\Lambda}_0} \leq \sigma^2_{\epsilon_i}\chi^2_d(1-\delta_i). \tag{17}$$

From (15), by Boole's inequality[4], we have that with probability at least $(1-2\delta_i)$, for all $t \geq 0$, and any $\mathbf{a} \in \mathbb{R}^d$,

$$|\mathbf{a}^T(\bar{\boldsymbol{k}} - \boldsymbol{k}^*)| \leq \|\mathbf{a}\|_{\mathbf{\Lambda}^{-1}}\, \sigma_{\epsilon_i}\left(\sqrt{2\log\left(\frac{1}{\delta_i}\frac{\det(\mathbf{\Lambda})^{1/2}}{\det(\mathbf{\Lambda}_0)^{1/2}}\right)} + \sqrt{\frac{\lambda_{\max}(\mathbf{\Lambda}_0)}{\lambda_{\min}(\mathbf{\Lambda})}\chi^2_d(1-\delta_i)}\right).$$

Let $\mathbf{a} = \mathbf{\Lambda}(\bar{\boldsymbol{k}} - \boldsymbol{k}^*)$ in the expression above, to obtain

$$\left\|\bar{\boldsymbol{k}} - \boldsymbol{k}^*\right\|^2_{\mathbf{\Lambda}} \leq \left\|\mathbf{\Lambda}(\bar{\boldsymbol{k}} - \boldsymbol{k}^*)\right\|_{\mathbf{\Lambda}^{-1}} \sigma_{\epsilon_i}\left(\sqrt{2\log\left(\frac{1}{\delta_i}\frac{\det(\mathbf{\Lambda})^{1/2}}{\det(\mathbf{\Lambda}_0)^{1/2}}\right)} + \sqrt{\frac{\lambda_{\max}(\mathbf{\Lambda}_0)}{\lambda_{\min}(\mathbf{\Lambda})}\chi^2_d(1-\delta_i)}\right).$$

Since $\left\|\mathbf{\Lambda}(\bar{\boldsymbol{k}} - \boldsymbol{k}^*)\right\|_{\mathbf{\Lambda}^{-1}} = \left\|\bar{\boldsymbol{k}}_t - \boldsymbol{k}^*\right\|_{\mathbf{\Lambda}}$, we divide both sides by $\left\|\bar{\boldsymbol{k}} - \boldsymbol{k}^*\right\|_{\mathbf{\Lambda}}$ and obtain

$$\left\|\bar{\boldsymbol{k}} - \boldsymbol{k}^*\right\|_{\mathbf{\Lambda}} \leq \sigma_{\epsilon_i}\left(\sqrt{2\log\left(\frac{1}{\delta_i}\frac{\det(\mathbf{\Lambda})^{1/2}}{\det(\mathbf{\Lambda}_0)^{1/2}}\right)} + \sqrt{\frac{\lambda_{\max}(\mathbf{\Lambda}_0)}{\lambda_{\min}(\mathbf{\Lambda})}\chi^2_d(1-\delta_i)}\right) \quad \forall t \geq 0, \tag{18}$$

which holds with probability at least $(1-2\delta_i)$. As this is the expression for β_i, this concludes our proof. □

B.2 Proof of Theorem 2: Probabilistic Safety

Next, we prove our result of probabilistic safety. First, we restate the theorem for ease of reading.

Theorem 2 (Probabilistic Safety)**.** *Compute confidence sets for model parameters using* (4) *and* (5)*. Using these confidence sets, compute probabilistic reachable sets* $\{\mathcal{X}^{t,\delta}_k\}^N_{k=1}$ *satisfying* (7)*. Using these sets, apply Algorithm 1 and sequentially solve (**Explore-OCP**) and (**Reach-OCP**).*

*Then, assuming that (**Reach-OCP**) is feasible at some time t, and under Assumptions 3 and 4, the system is guaranteed to satisfy* (1c) *i.e., to be safe at all times and eventually reach* $\mathcal{X}_{goal}$ *with probability* $(1-\delta)$*.*

Proof. We use a proof by construction. First, let N^j_{info}, and $t_j = \sum_{l=1}^{j-1} N^l_{\text{info}}$ be, respectively, the planning horizon, and the start time index of each $($**Explore-OCP**$)_j$, where $j = 1, \ldots, n_{\text{info}}$, with n_{info} the number of exploration phases. Similarly, define N_{reach}, and t_f to be, respectively, the planning horizon, and the start time index of (**Reach-OCP**). For conciseness, define $\boldsymbol{x}^{t_j}_k = \boldsymbol{x}_{t_j+k}$, corresponding to the state at time (t_j+k) in the j-th phase. Note that without feedback, open-loop controls satisfy $\boldsymbol{u}_k \in \mathcal{U}\ \forall k$. Further, define the event that the trajectory during the j-th exploration phase (or exploitation phase) satisfies all constraints as

$$\{\boldsymbol{x}^j_{\text{info}} \in \mathcal{X}^j_{\text{info}}\} = \left\{\bigwedge_{k=1}^{N^j_{\text{info}}} \left(\boldsymbol{x}^{t_j}_k \in \mathcal{X}_{\text{free}}\right) \cap \left(\boldsymbol{x}^{t_j}_{N^j_{\text{info}}} \in \mathcal{X}_0\right)\right\}, \quad j = 1, \ldots, n_{\text{info}}, \tag{19}$$

$$\{\boldsymbol{x}_{\text{reach}} \in \mathcal{X}_{\text{reach}}\} = \left\{\bigwedge_{k=1}^{N_{\text{reach}}} \left(\boldsymbol{x}^{t_f}_k \in \mathcal{X}_{\text{free}}\right) \cap \left(\boldsymbol{x}^{t_f}_{N_{\text{reach}}} \in \mathcal{X}_{\text{goal}}\right)\right\}. \tag{20}$$

[4] $\mathbb{P}((16) \cap (17)) = 1 - \mathbb{P}((16)^C \cup (17)^C)$, where A^C denotes the negation of A. Then, by Boole's inequality, $1 - \mathbb{P}((16)^C \cup (17)^C) \geq 1 - \mathbb{P}((16)^C) - \mathbb{P}((17)^C) = -1 + \mathbb{P}((16)) + \mathbb{P}((17))$. Finally, using the lower bounds on the probabilities that (16) and (17) occur, we obtain $\mathbb{P}((16) \cap (17)) \geq -1 + (1-\delta_i) + (1-\delta_i) = 1 - 2\delta_i$.

With this notation, we rewrite the safety condition of the original problem we are solving (which is the one we want to prove in this theorem) as

$$
\begin{aligned}
(1c) &= \mathbb{P}\Big(\bigwedge_{k=1}^{N_{\text{info}}^1} \big(\boldsymbol{x}_k \in \mathcal{X}_{\text{free}}\big) \cap \big(\boldsymbol{x}_{N_{\text{info}}^1} \in \mathcal{X}_0\big) \cap \cdots \cap \bigwedge_{k=\sum_i N_{\text{info}}^i}^{\sum_i N_{\text{info}}^i + N_{\text{reach}}} \big(\boldsymbol{x}_k \in \mathcal{X}_{\text{free}}\big) \cap \big(\boldsymbol{x}_N \in \mathcal{X}_{\text{goal}}\big)\Big) \\
&= \mathbb{P}\Big(\bigwedge_{k=1}^{N_{\text{info}}^1} \big(\boldsymbol{x}_k^{t_1} \in \mathcal{X}_{\text{free}}\big) \cap \big(\boldsymbol{x}_{N_{\text{info}}^1}^{t_1} \in \mathcal{X}_0\big) \cap \cdots \cap \bigwedge_{k=1}^{N_{\text{reach}}} \big(\boldsymbol{x}_k^{t_f} \in \mathcal{X}_{\text{free}}\big) \cap \big(\boldsymbol{x}_{N_{\text{reach}}}^{t_f} \in \mathcal{X}_{\text{goal}}\big)\Big) \\
&= \mathbb{P}\Big(\bigwedge_{j=1}^{n_{\text{info}}} \{\boldsymbol{x}_{\text{info}}^j \in \mathcal{X}_{\text{info}}^j\} \cap \{\boldsymbol{x}_{\text{reach}} \in \mathcal{X}_{\text{reach}}\}\Big) := \mathbb{P}\Big(\{\text{Safely Reached}\}\Big).
\end{aligned}
$$

Next, using the above, and by the law of total probability, we note that

$$
\begin{aligned}
(1c) = \mathbb{P}\Big(\{\text{Safely Reached}\} \,|\, \boldsymbol{k}_i^* \in \mathcal{C}_{i,t}^{\delta}\ \forall t,\ \forall i\Big) \cdot \mathbb{P}\Big(\boldsymbol{k}_i^* \in \mathcal{C}_{i,t}^{\delta}\ \forall t,\ \forall i\Big) + \\
\mathbb{P}\Big(\{\text{Safely Reached}\} \,|\, \boldsymbol{k}_i^* \notin \mathcal{C}_{i,t}^{\delta}\ \forall t,\ \forall i\Big) \cdot \mathbb{P}\Big(\boldsymbol{k}_i^* \notin \mathcal{C}_{i,t}^{\delta}\ \forall t,\ \forall i\Big) \\
\geq \mathbb{P}\Big(\{\text{Safely Reached}\} \,|\, \boldsymbol{k}_i^* \in \mathcal{C}_{i,t}^{\delta}\ \forall t,\ \forall i\Big) \cdot \mathbb{P}\Big(\boldsymbol{k}_i^* \in \mathcal{C}_{i,t}^{\delta}\ \forall t,\ \forall i\Big),
\end{aligned}
\tag{21}
$$

where $t = t_1, \ldots, t_{n_{\text{info}}}, t_f$, and $i = 1, \ldots, n$.

By Assumption 3, our meta-learning model can fit the true dynamics. Hence, if the true parameters are within the confidence sets $\mathcal{C}_{i,t}^{\delta}$, then, the reachable sets $\mathcal{X}_k^{t,\delta}$ necessarily contain the state trajectory on the true system, by definition (7). Using this fact, we can reformulate the constraints using the reachable sets, since

$$
\Big\{\mathcal{X}_k^{t,\delta} \subset \mathcal{X}_{\text{free}}\Big\} = \Big\{\boldsymbol{x}_k(\boldsymbol{K}^*) \in \mathcal{X}_{\text{free}} \,|\, \boldsymbol{k}_i^* \in \mathcal{C}_{i,t}^{\delta},\ \forall i\Big\}. \tag{22}
$$

By definition of (**Explore-OCP**) and (**Reach-OCP**), the reachable sets are subsets of the safe set, and the solution satisfies constraints. Hence, given a solution to these problems, we obtain

$$
\mathbb{P}\Big(\boldsymbol{x}_k^t(\boldsymbol{K}^*) \in \mathcal{X}_{\text{free}},\ k{=}1,\ldots,N \,|\, \boldsymbol{k}_i^* \in \mathcal{C}_{i,t}^{\delta},\ i{=}1,\ldots,n\Big) = \mathbb{P}\Big(\mathcal{X}_k^{t,\delta} \subset \mathcal{X}_{\text{free}}, k{=}1,\ldots,N\Big) = 1,
$$

which also holds for the final constraints $\boldsymbol{x}_N^t \in \mathcal{X}_0$, and $\boldsymbol{x}_N^{t_f} \in \mathcal{X}_{\text{goal}}$. Thus,

$$
\mathbb{P}\Big(\{\text{Safely Reached}\} \,|\, \boldsymbol{k}_i^* \in \mathcal{C}_{i,t}^{\delta}\ \forall t,\ \forall i\Big) = 1.
$$

Combining this result with (21), we obtain that

$$
(1c) \geq \mathbb{P}\Big(\boldsymbol{k}_i^* \in \mathcal{C}_{i,t}^{\delta}\ \forall t,\ \forall i\Big). \tag{23}
$$

This last term holds with probability greater than $(1 - \delta)$. Indeed, using (a) Boole's inequality, and (b) Theorem 1, we obtain

$$
\begin{aligned}
\mathbb{P}\Big(\boldsymbol{k}_i^* \in \mathcal{C}_{i,t}^{\delta}\ \forall t,\ \forall i\Big) &= \mathbb{P}\Big(\bigwedge_{i=1}^{n} \bigwedge_{t} \boldsymbol{k}_i^* \in \mathcal{C}_{i,t}^{\delta}\Big) = 1 - \mathbb{P}\Big(\bigvee_{i=1}^{n} \bigvee_{t} \boldsymbol{k}_i^* \notin \mathcal{C}_{i,t}^{\delta}\Big) \\
&\overset{(a)}{\geq} 1 - \sum_{i=1}^{n} \mathbb{P}\Big(\bigvee_{t} \boldsymbol{k}_i^* \notin \mathcal{C}_{i,t}^{\delta}\Big) = 1 - \sum_{i=1}^{n} \Big(1 - \mathbb{P}\Big(\bigwedge_{t} \boldsymbol{k}_i^* \in \mathcal{C}_{i,t}^{\delta}\Big)\Big) \\
&\overset{(b)}{\geq} 1 - \sum_{i=1}^{n} \big(1 - (1 - 2\delta_i)\big) = 1 - \sum_{i=1}^{n} \big(2\delta_i\big) = (1 - \delta).
\end{aligned}
$$

Since $\delta_i = \delta/(2n)$, combined with (23), this concludes this proof. □

B.3 Proof of Theorem 3: Probabilistic Feasibility

Theorem 3 (Probabilistic Feasibility)**.** *Under Assumptions 2, 3 and 4, there exists an optimization horizon N ensuring the feasibility of (**Explore-OCP**) at all times with probability* $(1-\delta)$.

Proof. Let n_{info} the number of exploration phases before (**Reach-OCP**) becomes feasible[5].

Also, let N^j_{info}, and $t_j = \sum_{l=0}^{j-1} N^l_{\text{info}}$ be, respectively, the planning horizon, and the start time index of each $(\textbf{Explore-OCP})_j$.

For conciseness, define $(\textbf{EOCP})_j$ for $\{(\textbf{Explore-OCP})_j$ is feasible$\}$, i.e., the event that the j-th exploration problem is feasible.

Then, by the law of total probability,

$$\begin{aligned}
\mathbb{P}\Big(\bigwedge_{j=0}^{n_{\text{info}}}(\textbf{EOCP})_j\Big) &= \mathbb{P}\Big(\bigwedge_{j=0}^{n_{\text{info}}}(\textbf{EOCP})_j,\ \boldsymbol{x}_{t_{n_{\text{info}}}} \in \mathcal{X}_0\Big) + \mathbb{P}\Big(\bigwedge_{j=0}^{n_{\text{info}}}(\textbf{EOCP})_j,\ \boldsymbol{x}_{t_{n_{\text{info}}}} \notin \mathcal{X}_0\Big) \\
&\geq \mathbb{P}\Big(\bigwedge_{j=0}^{n_{\text{info}}}(\textbf{EOCP})_j,\ \boldsymbol{x}_{t_{n_{\text{info}}}} \in \mathcal{X}_0,\ \Big) \\
&= \mathbb{P}\Big((\textbf{EOCP})_{n_{\text{info}}} \mid \bigwedge_{j=0}^{n_{\text{info}}-1}(\textbf{EOCP})_j,\ \boldsymbol{x}_{t_{n_{\text{info}}}} \in \mathcal{X}_0\Big)\mathbb{P}\Big(\bigwedge_{j=0}^{n_{\text{info}}-1}(\textbf{EOCP})_j,\ \boldsymbol{x}_{t_{n_{\text{info}}}} \in \mathcal{X}_0\Big).
\end{aligned}$$

By Assumption 2, given that $\boldsymbol{x}_{t_j} \in \mathcal{X}_0$, $(\textbf{Explore-OCP})_j$ is feasible for any j-th exploration phase. Indeed, choose $N^j_{\text{info}} = 1$ for $(\textbf{Explore-OCP})_j$. Then, $\boldsymbol{u}^j_0 = \boldsymbol{\pi}(\boldsymbol{x}_{t_j})$ is a feasible solution to $(\textbf{Explore-OCP})_j$. Thus, the event $\{(\textbf{EOCP})_j \mid \boldsymbol{x}_{t_j} \in \mathcal{X}_0\}$ holds with probability one.

In particular, $\{(\textbf{EOCP})_{n_{\text{info}}} \mid \bigwedge_{j=0}^{n_{\text{info}}-1}(\textbf{EOCP})_j, \boldsymbol{x}_{t_{n_{\text{info}}}} \in \mathcal{X}_0\}$ holds with probability one.

Next, we use the law of total probability to leverage our confidence sets over parameters:

$$\begin{aligned}
\mathbb{P}\Big(\bigwedge_{j=0}^{n_{\text{info}}}(\textbf{EOCP})_j\Big) &\geq \mathbb{P}\Big(\bigwedge_{j=0}^{n_{\text{info}}-1}(\textbf{EOCP})_j,\ \boldsymbol{x}_{t_{n_{\text{info}}}} \in \mathcal{X}_0\Big) \\
&\geq \mathbb{P}\Big(\bigwedge_{j=0}^{n_{\text{info}}-1}(\textbf{EOCP})_j,\ \boldsymbol{x}_{t_{n_{\text{info}}}} \in \mathcal{X}_0,\ \boldsymbol{k}^*_i \in \mathcal{C}^{\delta}_{i,t_{n_{\text{info}}-1}}\ \forall i\Big) \\
&= \mathbb{P}\Big(\boldsymbol{x}_{t_{n_{\text{info}}}} \in \mathcal{X}_0 \mid \bigwedge_j(\textbf{EOCP})_j,\ \boldsymbol{k}^*_i \in \mathcal{C}^{\delta}_{i,t_{n_{\text{info}}-1}} \forall i\Big)\,\mathbb{P}\Big(\bigwedge_j(\textbf{EOCP})_j,\ \boldsymbol{k}^*_i \in \mathcal{C}^{\delta}_{i,t_{n_{\text{info}}-1}} \forall i\Big).
\end{aligned}$$

By construction of the reachable sets $\{\mathcal{X}^{t_j,\delta}_k\}_{k=1}^{N^j_{\text{info}}}$, and by definition of $(\textbf{Explore-OCP})_j$ (since $\mathcal{X}^{t_j,\delta}_{N^j_{\text{info}}} \subset \mathcal{X}_0$), we have that $\boldsymbol{x}_{t_{j+1}} \in \mathcal{X}_0$ given that $(\textbf{Explore-OCP})_j$ is feasible and that $\boldsymbol{k}^*_i \in \mathcal{C}^{\delta}_{i,t_j}\ \forall i$, for any j-th exploration problem.

Thus, the first term $\{\boldsymbol{x}_{t_{n_{\text{info}}}} \in \mathcal{X}_0 \mid \bigwedge_j(\textbf{EOCP})_j,\ \boldsymbol{k}^*_i \in \mathcal{C}^{\delta}_{i,t_{n_{\text{info}}-1}} \forall i\}$ holds with probability one.

Thus,

$$\mathbb{P}\Big(\bigwedge_{j=0}^{n_{\text{info}}}(\textbf{EOCP})_j\Big) \geq \mathbb{P}\big(\bigwedge_{j=0}^{n_{\text{info}}-1}(\textbf{EOCP})_j,\ \boldsymbol{k}^*_i \in \mathcal{C}^{\delta}_{i,t_{n_{\text{info}}-1}}\ \forall i\big).$$

Since $(\textbf{EOCP})_0$ is feasible with probability one since $\boldsymbol{x}_0 \in \mathcal{X}_0$, and by reasoning by induction for all $j = n_{\text{info}}, \ldots, 0$, we obtain that

$$\mathbb{P}\Big(\bigwedge_{j=0}^{n_{\text{info}}}(\textbf{EOCP})_j\Big) \geq \mathbb{P}\big(\bigwedge_{j=0}^{n_{\text{info}}-1}\boldsymbol{k}^*_i \in \mathcal{C}^{\delta}_{i,t_j}\ \forall i\big) \geq (1-\delta),$$

where the last inequality comes from Theorem 1, which concludes this proof. □

[5] This result also holds if **(CC-OCP)** is not feasible, and the algorithm can never solve (**Reach-OCP**) to reach $\mathcal{X}_{\text{goal}}$ (e.g., if $\mathcal{X}_{\text{goal}}$ is surrounded by obstacles). Indeed, if the algorithm is stuck in an infinite number of exploration steps, the last inequality of this proof still holds for $n_{\text{info}} \to \infty$, by Theorem 1.

C Experimental Details and Further Results

Problem formulation and implementation: We evaluate our approach on a planar free-flying space robot. This system behaves (approximately) as a double integrator, controlled with gas thrusters and a reaction wheel. We consider the problem of cargo transport, in which the robot is attached to a payload that results in changes to the inertial properties of the system, resulting in nonlinear dynamics. This system mimics a cargo unloading scenario that is one plausible near-term application of autonomous robots on-board the International Space Station [46, 47].

The state of the system is given by $\boldsymbol{x} = [\mathbf{p}, \theta, \mathbf{v}, \omega] \in \mathbb{R}^6$, with $\mathbf{p}, \mathbf{v} \in \mathbb{R}^2$ the planar position and velocity, and $\theta, \omega \in \mathbb{R}$ the heading and angular velocity, respectively. For safety, we constrain $|v_i| \leq 0.2$ m/s, and $|\omega| \leq 0.25$ rad/s. The control inputs are $\boldsymbol{u} := [\mathbf{F}, M] \in \mathcal{U} \subset \mathbb{R}^3$, where $\mathcal{U} = [-\bar{u}_i, \bar{u}_i]$ represent the limited control authority from the gas thrusters. We set $\bar{u}_{1,2} = 0.15$ N, and $\bar{u}_3 = 0.01$ Nm. The payload causes a change in mass, inertia properties and causes the center of mass to be offset at $\mathbf{p}_0 \in \mathbb{R}^2$. The continuous time nonlinear dynamics of the system (which we write as $\dot{\boldsymbol{x}} = \mathbf{f}_t(\cdot)$) are

$$\dot{\mathbf{p}} = \mathbf{v},\ \ \dot{\theta} = \omega,\ \ \dot{\mathbf{v}} = \frac{1}{m}\left(\mathbf{F} - \dot{\omega}\begin{bmatrix} -p_{\text{o}y} \\ p_{\text{o}x} \end{bmatrix} + \omega^2 \mathbf{p}_{\text{o}}\right),\ \ \dot{\omega} = \frac{1}{J}\Big(M - p_{\text{o}x}F_y + p_{\text{o}y}F_x\Big). \tag{24}$$

We randomize the mass m, inertia J and center of mass offset $\mathbf{p}_0$ according to

$$m \sim \text{Unif}(25, 60)\,\text{kg},\ \ J \sim \text{Unif}(0.30, 0.70)\,\text{kg}\cdot\text{m}^2,\ \ p_{\text{o}i} \sim \text{Unif}(-7.5, 7.5)\,\text{cm},\ i \in \{x, y\}. \tag{25}$$

Using a zero-order hold on the controls and a forward Euler discretization scheme, we discretize (24) as

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Delta t \cdot \mathbf{f}_t(\boldsymbol{x}_k, \boldsymbol{u}_k, m, \mathbf{J}, \mathbf{p}_{\text{o}}) + \boldsymbol{\epsilon}_k, \tag{26}$$

where the $\epsilon_{k,i}$ are σ_{ϵ_i}-subgaussian, each bounded as $|\epsilon_{k,i}| \leq (\sigma_{\epsilon_i}^2 \chi_1^2(0.95))^{1/2}$. We use this discrete time nonlinear system in simulation, and to collect training data for offline meta-learning (see A.1).

We use a nominal model of the system $\boldsymbol{h}(\cdot,\cdot)$ using (24) with $(\bar{m}, \bar{J}, \bar{\mathbf{p}}_0) = (35, 0.4, \mathbf{0})$, which corresponds to a double integrator model. To represent the unknown model missmatch $\boldsymbol{g}(\cdot,\cdot,\boldsymbol{\theta})$, we train an ALPaCA model as described in [8] for 6000 iterations for all experiments.

For trajectory optimization, we use standard linear-quadratic final and step costs on states and controls to minimize control cost and deviation to $\mathcal{X}_0$ or $\mathcal{X}_{\text{goal}}$ depending on the phase. Specifically, we maximize the information cost defined in (12) while minimizing control effort, penalizing high velocities, and minizing the final distance to $\boldsymbol{x}_{\text{g}}$, the center of either $\mathcal{X}_0$, or $\mathcal{X}_{\text{goal}}$. We obtain

$$\max_{\boldsymbol{\mu},\boldsymbol{u}} \sum_{k=0}^{N-1}\Big(-\alpha_{\text{info}} l_{\text{info}}(\boldsymbol{\mu}_k, \boldsymbol{u}_k) + \boldsymbol{\mu}_k^T \boldsymbol{Q} \boldsymbol{\mu}_k + \boldsymbol{u}_k^T \boldsymbol{R} \boldsymbol{u}_k\Big) + (\boldsymbol{\mu}_N - \boldsymbol{x}_{\text{g}})^T \boldsymbol{Q}_N (\boldsymbol{\mu}_N - \boldsymbol{x}_{\text{g}}). \tag{27}$$

In these experiments, we set $\alpha_{\text{info}} = 0.025$ for exploration, whereas $\alpha_{\text{info}} = 0$ when reaching $\mathcal{X}_{\text{goal}}$, and $\boldsymbol{Q} = \text{diag}([0, 0, 0, 1, 1, 10])$, $\boldsymbol{R} = \text{diag}([10, 10, 10])$, and $\boldsymbol{Q}_N = 10^3\text{diag}([1, 1, 0.1, 10, 10, 10])$ for both (**Explore-OCP**) and (**Reach-OCP**).

Outline of results: We evaluate our framework on multiple problems (250) with different parameters $\boldsymbol{\theta}$. Specifically, we consider two different sets of σ_{ϵ_i}, and four environments with different boundary conditions and obstacles. For those problems, we evaluate the sensitivity to δ, to the number of samples M for reachability analysis, and the effect of the β-regularizer.

Sensitivity to the magnitude of $\boldsymbol{\epsilon}$: We consider two different noise levels:

1. $\sigma_{\epsilon_i}^2 = 10^{-7}$ for i=$1, 2, 4, 5$, and $\sigma_{\epsilon_i}^2 = 10^{-6}$ for i=$3, 6$.
2. $\sigma_{\epsilon_i}^2 = 10^{-6}$ for i=$1, 2$, $\sigma_{\epsilon_i}^2 = 10^{-5}$ for i=$3, 6$, and $\sigma_{\epsilon_i}^2 = 10^{-7}$ for i=$4, 5$.

Results for these different noise levels for different δ are reported in Tables 1, 2, and 3, where Table 1 is a subset of Table 2. From Table 2, we see that the performance and overall probability of safety for small noise levels is not sensitive to the chosen value of δ. We speculate that failures are mostly due to under-approximations from our approximate computation of the reachable sets with **randUP** [7]. For higher noise levels, it is evident that the conservatism of the algorithm can be tuned by choosing a different value for δ, since failures come from statistical errors from updating the model with noisy data (see Theorem 1). We also observe faster times to reach $\mathcal{X}_{\text{goal}}$ when opting for lower

small σ_{ϵ_i}	δ	# Explore	$\boldsymbol{x} \notin \mathcal{X}_{\text{obs}}$	$\boldsymbol{x} \in \mathcal{X}_{\text{min/max}}$	$\boldsymbol{x}_N \in \mathcal{X}_{\text{goal}}$	$\boldsymbol{x} \in \mathcal{X}_{\text{all}}$
SEELS	0.1	2.3 ± 0.01	$97.6 \pm 1.9\%$	$97.6 \pm 1.9\%$	$96.8 \pm 2.2\%$	$93.2 \pm 3.1\%$
SEELS	0.2	2.43 ± 0.19	$95.6 \pm 2.5\%$	$98.8 \pm 1.3\%$	$98.8 \pm 1.3\%$	$93.6 \pm 3.0\%$
SEELS	0.5	2.22 ± 0.18	$94.8 \pm 2.7\%$	$98.8 \pm 1.3\%$	$97.2 \pm 2.0\%$	$93.2 \pm 3.1\%$
Mean-Equivalent	-	0	$39.6 \pm 6.0\%$	$99.6 \pm 0.8\%$	$22.8 \pm 5.2\%$	$19.6 \pm 4.9\%$

Table 2: Results for 250 randomized experiments for different values of δ, with low noise levels $\boldsymbol{\epsilon}_k$, and $M = 1000$. For each experiment, we report the number of exploration phases, check constraints satisfaction, and report the percentage of experiments for which all constraints are satisfied $(\boldsymbol{x} \in \mathcal{X}_{\text{all}})$, with 95% confidence intervals. We run a mean-equivalent version of **SEELS** (Algorithm 1) which accounts for the disturbances $\boldsymbol{\epsilon}_k$, but does not consider model uncertainty. Our framework is guaranteed to simultaneously respect all constraints $(1 - \delta)$ fraction of the time, which is verified in practice.

high σ_{ϵ_i}	δ	# Explore	$\boldsymbol{x} \notin \mathcal{X}_{\text{obs}}$	$\boldsymbol{x} \in \mathcal{X}_{\text{min/max}}$	$\boldsymbol{x}_N \in \mathcal{X}_{\text{goal}}$	$\boldsymbol{x} \in \mathcal{X}_{\text{all}}$
SEELS	0.1	2.4 ± 0.14	$92.4 \pm 3.3\%$	$99.2 \pm 1.1\%$	$95.6 \pm 2.5\%$	$90.0 \pm 3.7\%$
SEELS	0.2	2.32 ± 0.13	$91.6 \pm 3.4\%$	$100 \pm 0\%$	$95.6 \pm 2.5\%$	$89.6 \pm 3.8\%$
SEELS	0.5	1.98 ± 0.11	$87.6 \pm 4.1\%$	$99.2 \pm 1.1\%$	$90.8 \pm 3.6\%$	$82.8 \pm 4.7\%$
Mean-Equivalent	-	0	$58.8 \pm 6.1\%$	$99.6 \pm 0.8\%$	$39.2 \pm 6.0\%$	$37.2 \pm 6.0\%$

Table 3: Results for 250 randomized experiments for different values of δ, with high noise levels $\boldsymbol{\epsilon}_k$, and $M = 2500$. Our safety guarantees are verified, and the need for exploration is evident, from the low success rate of an approach neglecting dynamics uncertainty.

probability of safety. In all scenarios, **SEELS** provides safety with high probability, verifying the theoretical guarantees of our framework.