arXiv:2404.15557v1 [cs.RO] 23 Apr 2024

Safe POMDP Online Planning among Dynamic Agents via Adaptive Conformal Prediction

Shili Sheng[1], Pian Yu[2], David Parker[2], Marta Kwiatkowska[2], and Lu Feng[1]

***Abstract*— Online planning for partially observable Markov decision processes (POMDPs) provides efficient techniques for robot decision-making under uncertainty. However, existing methods fall short of preventing safety violations in dynamic environments. This work presents a novel safe POMDP online planning approach that offers probabilistic safety guarantees amidst environments populated by multiple dynamic agents. Our approach utilizes data-driven trajectory prediction models of dynamic agents and applies Adaptive Conformal Prediction (ACP) for assessing the uncertainties in these predictions. Leveraging the obtained ACP-based trajectory predictions, our approach constructs safety shields on-the-fly to prevent unsafe actions within POMDP online planning. Through experimental evaluation in various dynamic environments using real-world pedestrian trajectory data, the proposed approach has been shown to effectively maintain probabilistic safety guarantees while accommodating up to hundreds of dynamic agents.**

I. Introduction

The *partially observable Markov decision process* (POMDP) framework is a general model for robot decision-making under uncertainty [1], which finds application in various robotic tasks, such as autonomous driving [2] and human-robot collaboration [3]. Significant progress in POMDP online planning, which interleaves policy computation and execution, has been made to overcome computational challenges. For instance, the widely adopted Partially Observable Monte Carlo Planning (POMCP) algorithm [4] enhances scalability through Monte Carlo sampling and simulation.

For many safety-critical robotic applications, computing POMDP policies that satisfy safety requirements is crucial. Existing methods for safe POMDP online planning often represent safety requirements as cost or chance constraints, aiming to maximize expected returns while reducing cumulative costs or failure probabilities [5], [6]. However, these methods cannot guarantee complete prevention of safety violations. In our previous work [7], we integrated POMCP with safety shields to ensure that, with probability one, goal states are reached and unsafe states are avoided. But these shielding methods are limited to static obstacles and fall short in dynamic environments.

*This work was supported in part by NSF grants CCF-1942836 and CCF-2131511, and the ERC ADG FUN2MODEL (Grant agreement ID: 834115).

[1]Shili Sheng and Lu Feng are with the School of Engineering and Applied Science, University of Virginia, Charlottesville, VA 22904, USA {ss7dr, lu.feng}@virginia.edu

[2]Pian Yu, David Parker, and Marta Kwiatkowska are with the Department of Computer Science, University of Oxford, Parks Road, Oxford OX1 3QD, United Kingdom {pian.yu, david.parker, marta.kwiatkowska}@cs.ox.ac.uk

To tackle this limitation, in this work we investigate safe POMDP online planning for a robotic agent travelling among multiple unknown dynamic agents, such as pedestrians or other robots. We consider a safety constraint, which specifies that the minimum distance from the robotic agent to any of the dynamic agents should exceed a predefined safety buffer. The goal is to develop a safe POMDP online planning method that computes an optimal policy maximizing the expected return while ensuring that the probability of satisfying the safety constraint exceeds a certain threshold.

This work addresses several key challenges. The first is the modeling of dynamic agents. We use data-driven trajectory models to predict the movements of these agents and apply Adaptive Conformal Prediction (ACP) to assess the uncertainties in these predictions, as per [8]. The second challenge involves the construction of safety shields that avert collisions with dynamic agents. To this end, we propose a novel algorithm that dynamically constructs safety shields to accommodate the ACP-estimated regions of dynamic agents. The third challenge is integrating safety shields into POMDP online planning. We enhance the POMCP algorithm with safety shields by evaluating the safety of each action during the Monte Carlo sampling and simulation process.

To the best of our knowledge, this is the first safe POMDP online planning approach that offers probabilistic safety guarantees in environments with dynamic agents. We evaluate the proposed approach through computational experiments in various dynamic environments, utilizing real-world pedestrian trajectory data.

A. Related Work

Safe POMDP online planning. Prior studies have explored different methods for incorporating safety constraints into online planning with POMDPs. Online algorithms for constrained POMDPs apply cost (or chance) constraints to limit expected cumulative costs (or failure probability); however, they do not guarantee the avoidance of constraint violations [5], [6]. An online method introduced in [9] synthesizes a partial conditional plan for POMDPs with a focus on safe reachability, aiming for specific probability thresholds to reach goals or avoid static obstacles. Additionally, a rule-based shielding method presented in [10] generates shields by learning parameters for expert-defined rule sets. In our previous work [7], we devised shields to preemptively prevent actions that could breach almost-sure reach-avoid specifications and incorporated these shields into the POMCP algorithm to ensure safe online planning for POMDPs. However, these methods primarily focus on circumventing static obstacles and are not directly applicable to safe planning in dynamic environments.

Planning among dynamic agents. Substantial research has been conducted in the field of robotic planning involving dynamic agents, such as pedestrians. Some works treat these dynamic agents as static obstacles, adapting to changes through online replanning [11]. Other approaches make simple assumptions about the dynamics of these agents; for example, they may assume that pedestrians move at a constant velocity [12]. There are also more sophisticated methods that model the intentions of dynamic agents. For instance, PORCA [13] is a POMDP-based planning method that accounts for complex models of pedestrians' intentions and interactions. Additionally, data-driven trajectory predictors have seen extensive use in existing works (e.g., [14], [15]). However, numerous data-driven prediction techniques, such as Long Short-Term Memory (LSTM), often lack mechanisms to convey uncertainty in their predictions, potentially resulting in decisions that compromise safety. In this work, we adopt LSTM-based trajectory predictors and enhance them by incorporating uncertainty estimation through conformal prediction.

Planning with conformal prediction. Conformal prediction offers techniques for estimating statistically rigorous uncertainty sets for predictive models, such as neural networks, without making assumptions about the underlying distributions or models [16]. Adaptive Conformal Prediction (ACP) extends these techniques to estimate prediction regions for time series data [17]. Recently, there has been a surge in integrating conformal prediction, including its adaptive variant, into planning frameworks to accommodate the uncertainty in predicted trajectories. For example, Model Predictive Control (MPC) based methods have been developed for safe planning in dynamic environments, which incorporate (adaptive) conformal prediction regions of the predicted trajectories of dynamic agents as constraints within the MPC framework [18], [8]. Furthermore, conformal prediction has been applied to quantify the uncertainty in trajectory predictions derived from diffusion dynamics models, aiding in planning and offline reinforcement learning applications [19]. Additionally, a hierarchical conformal natural language planning approach has been proposed to tackle motion planning challenges for mobile robots, particularly for tasks with temporal and logical constraints [20]. In the realm of large language models, conformal prediction has been leveraged to provide statistical guarantees on the completion of robot tasks, enhancing the reliability of language model-based planners [21].

In this work, we adopt the ACP-based trajectory predictor proposed in [8] and develop a safe POMDP online planning method that constructs shields on-the-fly incorporating ACP regions of dynamic agents.

II. Problem Formulation

POMDP model. We model the dynamics of a robotic agent as a POMDP, denoted as a tuple $\mathcal{M} = (S, A, O, T, R, Z, \gamma)$, where S, A and O are (finite) sets of states, actions, and observations, respectively; $T : S \times A \times S \to [0, 1]$ is the probabilistic transition function; $R : S \times A \to \mathbb{R}$ is the reward function; $Z : S \times A \times O \to [0, 1]$ is the observation function; and $\gamma \in [0, 1]$ is the discount factor. At each timestep t, the state $s_t \in S$ transitions to a successor state $s_{t+1} \in S$ with probability $T(s_t, a_t, s_{t+1}) = Pr(s_{t+1} \mid s_t, a_t)$ given an agent's action $a_t \in A$; the agent receives a reward $R(s_t, a_t)$, and makes an observation $o_{t+1} \in O$ about state s_{t+1} with probability $Z(s_{t+1}, a_t, o_{t+1}) = Pr(o_{t+1} \mid s_{t+1}, a_t)$.

Given the partial observability of POMDP states, the agent maintains a *history* of actions and observations, denoted by $h_t = a_0, o_1, \ldots, a_{t-1}, o_t$. A *belief state* represents the posterior probability distribution over states conditioned on the history, denoted by $b_t(s) = Pr(s_t = s|h_t)$ for $s \in S$. Let b_0 denote the initial belief state, representing a distribution over the POMDP's initial states. Let B denote the set of belief states of POMDP $\mathcal{M}$. The *belief support* of a belief state $b \in B$ is defined as $supp(b) := \{s \in S | b(s) > 0\}$, which is a set of states with positive belief. The set of belief supports of POMDP $\mathcal{M}$ is defined as $S_B := \{\Theta \subseteq S \mid \forall s, s' \in \Theta, \mathsf{obs}(s) = \mathsf{obs}(s')\}$, where $\mathsf{obs} : S \to 2^O$ is a function representing the set of possible observations for a state.

A POMDP *policy*, denoted by $\pi : B \to A$, is a mapping from belief states to actions. At timestep t, executing a policy π involves selecting an action $a_t = \pi(b_t)$ based on the current belief state b_t, and subsequently updating to belief state b_{t+1} after observing o_{t+1} according to Bayes' rule:

$$b_{t+1}(s') = \frac{Z(s', a_t, o_{t+1}) \sum_{s \in S} T(s, a_t, s') b_t(s)}{\eta(o_{t+1} \mid b, a)} \quad (1)$$

where $\eta(o_{t+1} \mid b, a)$ is a normalizing constant representing the prior probability of observing o_{t+1}.

Let $R(b_t, a_t) := \sum_{s \in S} R(s, a_t) b_t(s)$ denote the expected immediate reward of taking action a_t in belief state b_t. The expected return from following policy π starting at initial belief state b_0 is defined as:

$$V^\pi(b_0) := \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t R\left(b_t, \pi(b_t)\right) \mid b_0]. \quad (2)$$

Dynamic agents. Consider a robotic agent operating in an environment with N dynamic agents whose trajectories are *a priori* unknown. Let $X_t := (X_{t,1}, \ldots, X_{t,N})$ denote the joint agent state at timestep t, where $X_{t,i}$ represents the state of the i-th dynamic agent. Assume the agents' trajectories adhere to an unknown distribution $\mathcal{D}$. Let $X := (X_0, X_1, \ldots) \sim \mathcal{D}$ be a random trajectory sampled from this distribution. We represent the safety constraint, which requires the minimum distance from the robotic agent to any of the N dynamic agents to exceed a safety buffer $\epsilon \in \mathbb{R}^+$, through the following Lipschitz continuous constraint function:

$$c(s_t, X_t) := \min_{i \in \{1, \ldots, N\}} \|s_t - X_{t,i}\| - \epsilon. \quad (3)$$

Given a belief state b_t of the POMDP $\mathcal{M}$, we compute the probability of b_t satisfying the safety constraint by summing over the probabilities $b_t(s)$ for all states $s \in supp(b_t)$ that meet the condition $c(s, X_t) \geq 0$, denoted as:

$$\rho(b_t, X_t) := \sum_{s \in supp(b_t)} b_t(s) \cdot \mathbb{I}_{\{c(s,X_t) \geq 0\}}. \quad (4)$$

We define the expected average probability of satisfying the safety constraint across all potential POMDP executions initiated from the initial belief state b_0 under a policy π as:

$$\phi^\pi(b_0) := \mathbb{E}_\pi[\lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \rho(b_t, X_t) \mid b_0, X_t \sim \mathcal{D}]. \quad (5)$$

Problem. Given a POMDP model $\mathcal{M}$ for a robotic agent with initial belief state b_0, unknown random trajectories $X \sim \mathcal{D}$ of N dynamic agents, and a failure rate $\delta \in (0, 1)$, the objective is to compute an optimal POMDP policy π^* that maximizes the expected return $V^\pi(b_0)$ while ensuring that the expected average probability of satisfying the safety constraint is at least $1 - \delta$, denoted by $\phi^\pi(b_0) \geq 1 - \delta$.

III. Preliminaries

To tackle the problem under consideration, we propose a safe online planning approach for POMDPs that accounts for the uncertainty of dynamic agents' predicted trajectories. The key idea is as follows. Initially, we convert the probabilistic safety constraint concerning dynamic agents, i.e., $\phi^\pi(b_0) \geq 1 - \delta$, into an equivalent almost-sure safety constraint. This conversion is facilitated through Adaptive Conformal Prediction (ACP), a technique capable of adaptively quantifying the uncertainty of trajectory predictors and generating prediction regions with predefined probability thresholds. Subsequently, safety shields are constructed to accommodate the previously computed prediction regions of the dynamic agents, and they are integrated into the Partially Observable Monte Carlo Planning (POMCP) algorithm for online planning.

We now present the essential preliminaries on ACP for trajectory prediction in Section III-A, and the POMCP algorithm for online planning in Section III-B.

A. ACP-based Trajectory Prediction

Assume there exists a trajectory predictor capable of making predictions about the future trajectories of dynamic agents for a finite horizon H based on their past trajectories. Though our proposed approach is agnostic to the prediction method, we employ an LSTM model as the trajectory predictor in this work and make no additional assumptions about the trajectory distribution $\mathcal{D}$. To consider the uncertainty of predicted trajectories, which could influence the satisfaction of safety constraints, we compute ACP regions using the method described in [8].

Let $(\hat{X}_t^1, \ldots, \hat{X}_t^H)$ represent the predicted trajectory of dynamic agents' future states starting at timestep t and extending to the prediction horizon H, where $\hat{X}_t^\tau := (\hat{X}_{t,1}^\tau, \ldots, \hat{X}_{t,N}^\tau)$ is the predicted joint state of N agents made at timestep t for horizon $\tau \in \{1, \ldots, H\}$. However, we cannot evaluate the prediction error for future states since the ground truths $(X_{t+1}, \ldots, X_{t+H})$ are unknown at timestep t.

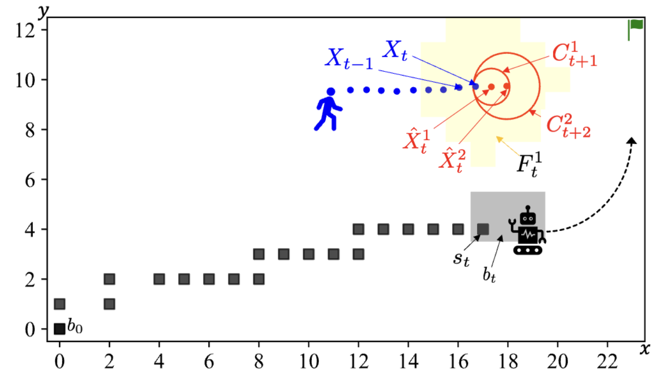

Fig. 1: Example gridworld with a robot navigating towards a flag while avoiding a pedestrian. It moves *east*, *south*, *west*, or *north*, reaching the adjacent grid cell with probability 0.1 or one cell further with probability 0.9. Gray shadow: robot's belief state b_t including state s_t. Red circles: ACP regions of uncertain predictions about pedestrian states. Yellow shadow: unsafe states per one-step prediction at timestep t.

We adopt the concept of *time-lagged nonconformity score* as defined in [8], which quantifies the τ step-ahead prediction error made τ timesteps ago, denoted by $\beta_t^\tau := \|X_t - \hat{X}_{t-\tau}^\tau\|$.

For each prediction horizon $\tau \in \{1, \ldots, H\}$, we calculate an ACP region $\beta_t^\tau \leq C_t^\tau$ based on $(\beta_{t-K}^\tau, \ldots, \beta_{t-1}^\tau)$ with a sliding window of size K, ensuring $Pr(\beta_t^\tau \leq C_t^\tau) \geq 1 - \delta$, where $\delta \in (0, 1)$ denotes a failure probability. The value of C_t^τ is determined by identifying the $\lceil (K + 1)(1 - \lambda_t^\tau) \rceil^{\text{th}}$ smallest value among $(\beta_{t-K}^\tau, \ldots, \beta_{t-1}^\tau)$, with the parameter λ_t^τ updated recursively as follows:

$$\lambda_t^\tau := \lambda_{t-1}^\tau + \alpha(\delta - \mathbb{I}_{\{C_{t-1}^\tau < \beta_{t-1}^\tau\}}) \quad (6)$$

where $\alpha \in (0, 1)$ is the learning rate and $\lambda_0^1 \in (0, 1)$ is a constant for the initial value.

Example 1. Consider a robot navigating in a gridworld with one dynamic agent (a pedestrian) as shown in Figure 1. Let $\langle x, y \rangle$ denote a two-dimensional position in the gridworld. At timestep t, the dynamic agent's (joint) state is $X_t = (\langle 16.702, 9.726 \rangle)$. A trajectory predictor with the prediction horizon $H = 2$ yields one-step and two-step ahead predictions as $\hat{X}_t^1 = (\langle 17.334, 9.711 \rangle)$ and $\hat{X}_t^2 = (\langle 17.947, 9.743 \rangle)$, respectively. We compute the time-lagged nonconformity score $\beta_t^1 = \|X_t - \hat{X}_{t-1}^1\| = 0.068$, where $\hat{X}_{t-1}^1 = (\langle 16.650, 9.682 \rangle)$ is the prediction about state X_t made one timestep ago at $t - 1$.

Next we compute the ACP region C_{t+1}^1 to ensure $Pr(\beta_{t+1}^1 \leq C_{t+1}^1) \geq 1 - \delta$ such that the prediction error $\beta_{t+1}^1 = \|X_{t+1} - \hat{X}_t^1\|$ is bounded with the failure probability $\delta = 0.05$. Let the size of sliding window be $K = 30$ and the learning rate $\alpha = 0.0008$. Suppose $C_t^1 = 0.736$ and $\lambda_t^1 = 0.0495$. We have $\lambda_{t+1}^1 = \lambda_t^1 + \alpha(\delta - \mathbb{I}_{\{C_t^1 < \beta_t^1\}}) = 0.04954$. We determine the value of C_{t+1}^1 by finding the $\lceil (K + 1)(1 - \lambda_{t+1}^1) \rceil^{\text{th}}$, which is the 30^{th} smallest value among $(\beta_{t-29}^1, \ldots, \beta_t^1)$. This process yields $C_{t+1}^1 = 0.736$. Similarly, the ACP region to bound the prediction error

$\beta^2_{t+2} = \|X_{t+2} - \hat{X}^2_t\|$ is computed as $C^2_{t+2} = 1.329$. The obtained ACP regions are plotted as red circles in Figure 1, centered at $\hat{X}^1_t$ and $\hat{X}^2_t$, with radii C^1_{t+1} and C^2_{t+2}, respectively. ■

B. Partially Observable Monte-Carlo Planning

We employ the *Partially Observable Monte Carlo Planning* (POMCP) algorithm [4], a popular method for POMDP online planning that interleaves the policy computation and execution. Each timestep t starts with the POMCP algorithm deploying Monte Carlo tree search [22] to navigate a search tree. The root node of this tree is $\mathcal{T}(h_t) = \langle \mathcal{N}(h_t), \mathcal{V}(h_t), \mathcal{B}(h_t) \rangle$. Here, $\mathcal{N}(h_t)$ counts how often the history h_t has been visited, $\mathcal{V}(h_t)$ calculates the expected return from all simulations starting at h_t, and $\mathcal{B}(h_t)$ contains particles representing POMDP states that estimate the belief state b_t. The algorithm iterates through four main steps:

(1) **Selection:** A state s is selected at random from the particle set $\mathcal{B}(h_t)$. (2) **Expansion:** Upon reaching a leaf node $\mathcal{T}(h)$, it introduces new child nodes for every action $a \in A$, expressed as $\mathcal{T}(ha) = \langle \mathcal{N}_{init}(ha), \mathcal{V}_{init}(ha), \emptyset \rangle$. (3) **Simulation:** If $\mathcal{T}(h)$ is a non-leaf node, an action a is selected to maximize $\mathcal{V}(ha) + c\sqrt{\frac{\log \mathcal{N}(h)}{\mathcal{N}(ha)}}$ using the *upper confidence bound* (UCB) to balance exploration and exploitation. In leaf nodes, a is selected based on a predefined rollout policy like uniform random selection. Then, the next state s' is simulated using a black-box simulator $(s', o, r) \sim \mathcal{G}(s, a)$ and added to $\mathcal{B}(hao)$. This process runs until reaching a predetermined depth. (4) **Backpropagation:** After simulation, the search tree nodes are updated with new data from the path.

The planning phase at timestep t concludes once a target number of these iterations has occurred or a time limit is reached. The agent then takes the best action $a_t = \arg\max_a \mathcal{V}(h_t a)$, receives a new observation o_{t+1}, and proceeds to the next step, constructing a search tree from the new root node $\mathcal{T}(h_t a_t o_{t+1})$. POMCP is effective because it mitigates the *curse of dimensionality* through state sampling and the *curse of history* via history sampling with a black-box simulator.

IV. Approach

We develop a novel approach that constructs safety shields on-the-fly using ACP-based trajectory predictions for dynamic agents, and shields unsafe actions in POMDP online planning. We first define safety shields in Section IV-A, present an algorithm for on-the-fly shield construction in Section IV-B, describe the shielding method for safe online planning in Section IV-C, and analyze the correctness and complexity of the proposed approach in Section IV-D.

A. ACP-induced Safety Shield

Given a τ-step ahead prediction $\hat{X}^\tau_t$ made at timestep t for the dynamic agents' future state, and its corresponding ACP region $C^\tau_{t+\tau}$ computed as per Section III-A, we define the safety constraint using the Lipschitz continuous function described in Equation 3 as follows:

$$c(s_{t+\tau}, \hat{X}^\tau_t) \geq L \cdot C^\tau_{t+\tau} \tag{7}$$

where $L > 0$ is the Lipschitz constant and $\tau \in \{1, \dots, H\}$ with the prediction horizon H.

Denote by $F^\tau_t := \{s \in S \mid c(s, \hat{X}^\tau_t) < L \cdot C^\tau_{t+\tau}\}$ the τ-step ahead prediction region of $\hat{X}^\tau_t$ made at timestep t. It follows from [8] that $Pr(X_{t+\tau} \in F^\tau_t) \geq 1 - \delta, \forall \tau \in \{1, \dots, H\}$, where $X_{t+\tau}$ is the true state of the dynamic agents at timestep $t + \tau$. By ensuring that the prediction regions $\{F^\tau_t\}_{\tau \in \{1,\dots,H\}}$ are avoided almost-surely (i.e., with probability 1) at every timestep t, it becomes possible to achieve the desired probabilistic safety guarantee, i.e., $\phi^\pi(b_0) \geq 1 - \delta$, with regards to the dynamic agents.

Example 2. We have $\hat{X}^1_t = (\langle 17.334, 9.711 \rangle)$ and $C^1_t = 0.736$ from Example 1. Let the Lipschitz constant be $L = 1$ and the safety buffer $\epsilon = 2$. Suppose the robot's actual state at timestep $t+1$ is $s_{t+1} = (18, 4)$. We have $c(s_{t+1}, \hat{X}^1_t) = \|s_{t+1} - \hat{X}^1_t\| - \epsilon = 3.7497 \geq C^1_t$. Thus, the safety constraint is satisfied. ■

In [23], it was demonstrated that, for enforcing almost-sure safety specifications, belief probabilities are irrelevant and only the belief support is important. Inspired by this observation, we define a winning belief support and winning regions for a given horizon h, which are used for shielding unsafe actions during online planning.

We say that a POMDP policy π is *winning* from belief state b_t for a finite horizon $h \leq H$ iff every state in the belief supports $supp(b_{t+j})$ for $j \in \{0, \dots, h\}$ of all possible executions under the policy π satisfies the safety constraint of Equation 7.

A belief state $b \in B$ is considered winning for a horizon h if there exists an h-step horizon winning policy π originating from b, and the belief support, denoted as $supp(b)$, is termed a *winning belief support* for the horizon h. A set of belief supports, denoted by $W \subseteq S_B$, is termed a *winning region* for an h-step horizon if every belief support $\Theta \in W$ is winning for the horizon h. In the special case when $h = 0$, we say that W is a winning region with a zero-step horizon iff all states belonging to each belief support $supp(b) \in W$ satisfy the safety constraint of Equation 7.

To enforce safety, we can define a safety shield, denoted by $\xi : \Theta \to 2^A$, which restricts actions to those leading solely to successor belief supports within the winning region W.

B. Computing Winning Regions for Shields

We present Algorithm 1 for computing a set of winning regions, denoted by $\{W^\tau_t\}^H_{\tau=1}$, to construct safety shields on-the-fly at timestep t, where each winning region W^τ_t has a winning horizon of $H - \tau$.

First, given a winning belief state b_t, we compute the set of belief supports of the POMDP $\mathcal{M}$ that are reachable within H steps, denoted by $S^{b_t,H}_B \subseteq S_B$.

Next, we construct the reachable fragment of a belief-support transition system (BSTS) for the POMDP $\mathcal{M}$, denoted by a tuple $\mathcal{M}^{b_t,H}_B = \{S^{b_t,H}_B \times Q, \bar{s}^{b_t,H}_B, A, T^{b_t,H}_B\}$, where the state space is the product of $S^{b_t,H}_B$ and a time counter $Q = \{0, 1, \dots, H\}$, the initial state is $\bar{s}^{b_t,H}_B = \langle supp(b_t), 0 \rangle$, the set of actions A are the

Algorithm 1: Computing winning regions

Input: POMDP model $\mathcal{M}$, a winning belief state b_t, a set of dynamic agents' predicted states $\{\hat{X}_t^\tau\}_{\tau=1}^H$ and ACP regions $\{C_{t+\tau}^\tau\}_{\tau=1}^H$.
Output: A set of winning regions $\{W_t^\tau\}_{\tau=1}^H$.

Compute the set of reachable belief supports $S_B^{b_t,H}$
Construct a belief-support transition system $\mathcal{M}_B^{b_t,H}$
Compute the set of unsafe states Ψ_t in $\mathcal{M}_B^{b_t,H}$
$W_t^H \leftarrow S_B^{b_t,H} \setminus \{\Theta \in S_B^{b_t,H} \mid \langle \Theta, H\rangle \in \Psi_t\}$
for $\tau = H-1$ **to** 1 **do**
    foreach $\Theta \in S_B^{b_t,H}$ **do**
        if $\langle \Theta, \tau\rangle \notin \Psi_t$ **then**
            foreach $a \in A$ **do**
                if $\mathrm{post}(\langle \Theta, \tau\rangle, a) \subseteq W_t^{\tau+1}$ **then**
                    insert Θ to W_t^τ
return $\{W_t^\tau\}_{\tau=1}^H$

Algorithm 2: Safe online planning via shielding

Input: POMDP model $\mathcal{M}$, an initial belief state b_0, a distribution $\mathcal{D}$ of dynamic agents' trajectories, a prediction horizon H, and a failure probability δ.
Output: A safe POMDP policy π^*.

1 **for** $t = 0$ **to** ∞ **do**
2   predict dynamic agents' trajectories $\{\hat{X}_t^\tau\}_{\tau=1}^H \sim \mathcal{D}$
3   compute ACP regions $\{C_{t+\tau}^\tau\}_{\tau=1}^H$ w.r.t failure rate δ
4   compute winning regions $\{W_t^\tau\}_{\tau=1}^H$ `// Algorithm 1`
5     $\pi^*(b_t) \leftarrow \mathsf{shieldPOMCP}(\mathcal{T}(h_t), \{W_t^\tau\}_{\tau=1}^H)$
6     $(o_{t+1}, b_{t+1}) \leftarrow$ execute action $\pi^*(b_t)$
7 **return** π^*

same as in the POMDP $\mathcal{M}$, and the transition function is given by $T_B^{b_t,H}(\langle\Theta, q\rangle, a) = \langle \Theta', q+1\rangle$ if $\Theta' \in \left\{\bigcup_{s\in\Theta}\{s' \in S \mid T(s,a,s') > 0 \text{ and } o \in \mathsf{obs}(s')\} \mid o \in O\right\}$. The transition function $T_B^{b_t,H}$ is constructed as follows. Given a belief support $\Theta \in S_B^{b_t,H}$, a timestep $q \in Q$, and an action a, our initial step involves calculating the set of all potential successor states from Θ and a. This is accomplished by determining the set of all possible successor states from each state $s \in \Theta$ and then taking the union of these sets. Following that, we reorganize these states into a set of belief supports based on the principle that states within a belief support share the same observation. Finally, the timestep is increased by 1. Let $\mathrm{post}(\langle\Theta, q\rangle, a) := \{\Theta' \mid T_B^{b_t,H}(\langle\Theta, q\rangle, a) = \langle\Theta', q+1\rangle\}$ denote the set of all possible successor belief supports.

Given a set of dynamic agents' predicted states $\{\hat{X}_t^\tau\}_{\tau=1}^H$ and ACP regions $\{C_{t+\tau}^\tau\}_{\tau=1}^H$, we compute all possible POMDP states where the safety constraint could be violated, denoted as $F_t^\tau := \{s \in S \mid c(s, \hat{X}_t^\tau) < L \cdot C_{t+\tau}^\tau\}$. The set of unsafe states in the BSTS $\mathcal{M}_B^{b_t,H}$ is defined as $\Psi_t := \{\langle\Theta, q\rangle \in S_B^{b_t,H} \times Q \mid \exists s \in \Theta \text{ such that } s \in F_t^q\}$.

We compute a set of winning regions $\{W_t^\tau\}_{\tau=1}^H$ recursively in a backward manner. Let W_t^H be the set of reachable belief supports $\Theta \in S_B^{b_t,H}$, excluding those that result in unsafe states where $\langle\Theta, H\rangle \in \Psi_t$. Starting from $\tau = H-1$, we add a reachable belief support $\Theta \in S_B^{b_t,H}$ to the winning region W_t^τ only if both of the following two conditions hold: (C1) $\langle\Theta, \tau\rangle$ does not belong to the unsafe set Ψ_t; and (C2) there exists an action $a \in A$ that leads solely to winning successor belief supports $\mathrm{post}(\langle\Theta, \tau\rangle, a) \subseteq W_t^{\tau+1}$ from Θ in the BSTS $\mathcal{M}_B^{b_t,H}$.

Example 3. Following previous examples, we compute the set of unsafe POMDP states for a one-step prediction at timestep t as $F_t^1 = \{s \in S \mid c(s, \hat{X}_t^1) < C_{t+1}^1\}$. These states are represented as the yellow shadow in Figure 1. Similarly, for a two-step prediction at timestep t, we compute the set of unsafe states as $F_t^2 = \{s \in S \mid c(s, \hat{X}_t^2) < C_{t+2}^2\}$. The winning region W_t^2 is identified as the set of belief supports $S_B^{b_t,2}$ that can be reached from b_t within two steps, while excluding those that contain any unsafe states from F_t^2. We say that W_t^2 has a zero-step winning horizon, because we do not evaluate the safety of actions leading to states beyond the prediction horizon $H = 2$. The one-step horizon winning region W_t^1 is identified as the set of reachable belief supports that do not contain unsafe states from F_t^1 and can lead solely to successor belief supports in W_t^2. ■

C. Safe Online Planning via Shielding

Algorithm 2 illustrates the proposed safe POMDP online planning approach. At each planning step t, it first predicts dynamic agents' trajectories $\{\hat{X}_t^\tau\}_{\tau=1}^H$ and computes ACP regions $\{C_{t+\tau}^\tau\}_{\tau=1}^H$ as described in Section III-A. Then, Algorithm 1 is applied to compute a set of winning regions $\{W_t^\tau\}_{\tau=1}^H$ for the safety shield.

In Line 5, the procedure $\mathsf{shieldPOMCP}$ is called, integrating shields with the POMCP algorithm (see Section III-B), which navigates a search tree whose root node is $\mathcal{T}(h_t)$. During the simulation phase of the POMCP algorithm, when an action $a \in A$ is selected (either by the UCB rule or during rollout) for the history $h_{t+\tau-1}$, with $\tau \in \{1, \ldots, H\}$, and a black-box simulator generates $(s', o, r) \sim \mathcal{G}(s, a)$, the procedure checks if the particle set $\mathcal{B}(h_{t+\tau-1}ao) \cup \{s'\}$ belongs to the winning region W_t^τ. If the resulting particle set is not a winning belief support, the branch of the tree starting from node $\mathcal{T}(h_{t+\tau-1}a)$ is pruned, effectively shielding action a at node $\mathcal{T}(h_{t+\tau-1})$. If the simulation depth exceeds H, no shield is applied to actions selected for any history beyond h_{t+H}.

When the POMCP planning concludes at timestep t, the best action a_t is selected from the set of allowed actions at node $\mathcal{T}(h_t)$ as the one that achieves the maximum value of $\mathcal{V}(h_t a)$. We set the policy π^* with $\pi^*(b_t) = a_t$. The agent executes a_t, receives an observation o_{t+1} and updates the belief state b_{t+1} for the next step.

Example 4. At timestep t, consider a state $s = \langle 17, 5\rangle$ sampled from the particle set $\mathcal{B}(h_t)$ at the root node $\mathcal{T}(h_t)$. During the POMCP simulation, suppose action $a =$ *east* is chosen for history h_t, and a black-box simulator yields $(s', o, r) \sim \mathcal{G}(s, a)$ with $s' = \langle 18, 5\rangle$. We then check if $\mathcal{B}(h_t ao) \cup \{s'\}$ falls within the winning region W_t^1; it does,

(a) ETH (b) Hotel (c) GC

Fig. 2: Example scenes of real-world pedestrians trajectories from benchmark datasets [24].

so the simulation process advances with action a' for history $h_t ao$. Assuming $a' = north$ leads to $(s'', o', r') \sim \mathcal{G}(s', a')$ with $s'' = \langle 18, 7 \rangle$, an unsafe state in F_t^2, the updated particle set $\mathcal{B}(h_t aoa'o') \cup \{s''\}$ falls outside of W_t^2. Therefore, action *north* is shielded at node $\mathcal{T}(h_t ao)$. ■

D. Correctness and Complexity

Correctness. The correctness of Algorithm 1 is stated in Lemma 1 and the correctness of Algorithm 2, with respect to the problem statement in Section II, is stated in Theorem 1. The proofs are given in the appendix.

Lemma 1: The output of Algorithm 1, denoted by $\{W_t^\tau\}_{\tau=1}^H$, comprises a set of winning regions, with each W_t^τ representing a winning region for an $(H - \tau)$-step horizon.

Theorem 1: Given a POMDP model $\mathcal{M}$ for a robotic agent with initial belief state b_0, the unknown random trajectories $X \sim \mathcal{D}$ of N dynamic agents with a prediction horizon H, and a failure probability $\delta \in (0, 1)$, the policy π^* computed by Algorithm 2 achieves the maximal expected return $V^{\pi^*}(b_0)$ while ensuring safety, i.e., $\phi^{\pi^*}(b_0) \geq 1 - \delta$.

Complexity. There are several components in the complexity analysis of the proposed approach in Algorithm 2. The complexity of predicting dynamic agents' trajectories depends on the underlying prediction model. The complexity of computing ACP regions at timestep t is $\mathcal{O}(H \cdot N \cdot K \cdot \log(K))$, depending on the prediction horizon H, the number of agents N, and the sliding window size K. The complexity of Algorithm 1 for computing winning regions to construct shields is bounded by $\mathcal{O}(H \cdot |S_B^{b_t, H}| \cdot |A|)$, which depends on the prediction horizon H, the number of H-step reachable belief supports $S_B^{b_t, H}$ from belief state b_t, and the size of the action set A. The overhead of adding shielding to POMCP, specifically checking against safety shields, is bounded by $\mathcal{O}\left((|A| \cdot |O|)^H\right)$ per simulation. In practice, the overhead is negligible, as demonstrated by the experimental results in the next section.

V. Experiments

We implemented the proposed approach and evaluated it through various computational experiments. All experiments were run on a MacBook Pro machine with 10-core 3.2 GHz Apple M1 processor and 16 GB of memory.

Environments. We consider three gridworld environments, where the movements of N dynamic agents follow real-world pedestrians trajectories derived from benchmark datasets [24]. Specifically, we utilize three datasets: ETH, Hotel, and GC, for which example scenes are illustrated in Figure 2. In each gridworld environment, the robot aims to reach a target destination while avoiding pedestrians. The robot can move *east*, *south*, *west*, or *north*, reaching the adjacent cell with probability 0.1 or one cell further with probability 0.9. The robot has partial observability of the environment due to noisy sensors, such that every four neighboring grid cells in a 2×2 block share the same observation. The reward function is defined as: $1,000$ for reaching the destination, -1 per step, and -10 per collision.

Hyperparameters. Each pedestrian trajectory dataset is split into training, validation, and test sets in a ratio of 16:4:5. LSTM models with 64 hidden units are trained for the trajectory prediction with a prediction horizon $H = 3$. We set the failure probability $\delta = 0.05$ and the safety buffer to be $\epsilon = 0.5$ grid. The ACP regions are computed with a learning rate $\alpha = 0.0008$. The POMCP algorithm is set with the following hyperparameters: the number of simulations is 4,096; the simulation depth is 200; and the number of particles sampled from the initial state distribution is 10,000.

Results. We compare the performance of the proposed approach for safe online planning via ACP-induced shields with two baselines: (i) *No Shield*, i.e., POMCP without shielding; and (ii) *Shielding without ACP*, i.e., POMCP with shields that do not account for ACP regions, by modifying the safety constraint in Equation 7 to $c(s_{t+\tau}, \hat{X}_t^\tau) \geq 0$.

Table I shows the results averaging over 100 runs of each method for different environments with varying number N of dynamic agents. Across all cases, the proposed approach achieves a better safety rate (i.e., measured by the percentage of time satisfying the safety constraint during a run) than the baselines. All three methods result in comparable travel times for the robot to reach the destination in the ETH and Hotel environments, while shielding approaches lead to longer travel times when the robot needs to avoid a significantly higher number of pedestrians in the GC environment. Table I also reports the mean and standard deviation of the minimum distance between the robot and pedestrians; the

TABLE I: Experiment Results

	ETH				Hotel				GC			
Method	N	Safety Rate	Time (s)	Min Distance	N	Safety Rate	Time (s)	Min Distance	N	Safety Rate	Time (s)	Min Distance
No Shield		0.893	21.1	0.28±0.19		0.944	20.1	0.42±0.21		0.91	39.3	0.22±0.13
Shielding without ACP	45	0.943	21.5	0.39±0.29	35	0.969	20.3	0.54±0.3	160	0.943	67.2	0.23±0.13
Shielding with ACP		**0.974**	22.1	0.51±0.29		**0.988**	20.6	0.8±0.49		**0.963**	71.4	0.28±0.16
No Shield		0.891	21.0	0.26±0.17		0.931	20.1	0.38±0.24		0.904	39.9	0.2±0.1
Shielding without ACP	55	0.951	21.8	0.41±0.25	45	0.959	20.1	0.48±0.24	180	0.938	66.8	0.23±0.12
Shielding with ACP		**0.975**	22.4	0.53±0.37		**0.982**	20.6	0.62±0.27		**0.953**	71.1	0.24±0.16
No Shield		0.872	21.2	0.24±0.13		0.921	20.2	0.36±0.18		0.895	39.6	0.22±0.11
Shielding without ACP	65	0.943	21.9	0.36±0.2	55	0.957	20.3	0.48±0.29	200	0.931	65.7	0.2±0.13
Shielding with ACP		**0.967**	22.6	0.42±0.26		**0.982**	20.3	0.6±0.24		**0.951**	74.3	0.25±0.15

proposed approach is more conservative than the baselines, maintaining a larger minimum distance for safety.

Finally, we observed that shielding does not significantly add to the runtime of online planning. The average computation time for each planning step of POMCP without shields is 0.28 seconds, compared to 0.35 seconds for the proposed approach, incurring an additional 0.07 seconds for constructing winning regions and shielding actions per step.

VI. Conclusion

This work developed a novel shielding approach aimed at ensuring safe POMDP online planning in dynamic environments with multiple unknown dynamic agents. We proposed to leverage ACP for predicting the future trajectories of these dynamic agents. Subsequently, safety shields are computed based on the prediction regions. Finally, we integrated these safety shields into the POMCP algorithm to enable safe POMDP online planning. Experimental results conducted on three benchmark domains, each with varying numbers of dynamic agents, demonstrated that the proposed approach successfully met the safety requirements while adhering to a predefined failure rate.

For future work, we will assess the effectiveness of the proposed approach across diverse POMDP domains and deploy it in real-world robotic tasks. Another important direction is to explore methods for handling continuous POMDPs to enhance scalability and generalization.

References

[1] M. Lauri, D. Hsu, and J. Pajarinen, “Partially observable markov decision processes in robotics: A survey,” *IEEE Transactions on Robotics*, 2022.

[2] S. Sheng, E. Pakdamanian, K. Han, Z. Wang, J. Lenneman, D. Parker, and L. Feng, “Planning for automated vehicles with human trust,” *ACM Transactions on Cyber-Physical Systems*, vol. 6, no. 4, pp. 1–21, 2022.

[3] P. Yu, S. Dong, S. Sheng, L. Feng, and M. Kwiatkowska, “Trust-aware motion planning for human-robot collaboration under distribution temporal logic specifications,” in *International Conference on Robotics and Automation (ICRA)*, 2024.

[4] D. Silver and J. Veness, “Monte-carlo planning in large pomdps,” *Advances in neural information processing systems*, vol. 23, 2010.

[5] J. Lee, G.-H. Kim, P. Poupart, and K.-E. Kim, “Monte-carlo tree search for constrained pomdps,” *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[6] M. Khonji, A. Jasour, and B. C. Williams, “Approximability of constant-horizon constrained pomdp.” in *IJCAI*, 2019, pp. 5583–5590.

[7] S. Sheng, D. Parker, and L. Feng, “Safe pomdp online planning via shielding,” in *International Conference on Robotics and Automation (ICRA)*, 2024.

[8] A. Dixit, L. Lindemann, S. X. Wei, M. Cleaveland, G. J. Pappas, and J. W. Burdick, “Adaptive conformal prediction for motion planning among dynamic agents,” in *Learning for Dynamics and Control Conference*. PMLR, 2023, pp. 300–314.

[9] Y. Wang, A. A. R. Newaz, J. D. Hernández, S. Chaudhuri, and L. E. Kavraki, “Online partial conditional plan synthesis for pomdps with safe-reachability objectives: Methods and experiments,” *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 3, pp. 932–945, 2021.

[10] G. Mazzi, A. Castellini, and A. Farinelli, “Risk-aware shielding of partially observable monte carlo planning policies,” *Artificial Intelligence*, vol. 324, p. 103987, 2023.

[11] A. Bauer, K. Klasing, G. Lidoris, Q. Mühlbauer, F. Rohrmüller, S. Sosnowski, T. Xu, K. Kühnlenz, D. Wollherr, and M. Buss, “The autonomous city explorer: Towards natural human-robot interaction in urban environments,” *International journal of social robotics*, vol. 1, pp. 127–140, 2009.

[12] R. Kümmerle, M. Ruhnke, B. Steder, C. Stachniss, and W. Burgard, “Autonomous robot navigation in highly populated pedestrian zones,” *Journal of Field Robotics*, vol. 32, no. 4, pp. 565–589, 2015.

[13] Y. Luo, P. Cai, A. Bera, D. Hsu, W. S. Lee, and D. Manocha, “Porca: Modeling and planning for autonomous driving among many pedestrians,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3418–3425, 2018.

[14] E. L. Zhu, F. L. Busch, J. Johnson, and F. Borrelli, “A gaussian process model for opponent prediction in autonomous racing,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 8186–8191.

[15] A. Farid, S. Veer, B. Ivanovic, K. Leung, and M. Pavone, “Task-relevant failure detection for trajectory predictors in autonomous vehicles,” in *Conference on Robot Learning*. PMLR, 2023, pp. 1959–1969.

[16] A. N. Angelopoulos, S. Bates *et al.*, “Conformal prediction: A gentle introduction,” *Foundations and Trends® in Machine Learning*, vol. 16, no. 4, pp. 494–591, 2023.

[17] M. Zaffran, O. Féron, Y. Goude, J. Josse, and A. Dieuleveut, “Adaptive conformal predictions for time series,” in *International Conference on Machine Learning*. PMLR, 2022, pp. 25 834–25 866.

[18] L. Lindemann, M. Cleaveland, G. Shim, and G. J. Pappas, “Safe planning in dynamic environments using conformal prediction,” *IEEE Robotics and Automation Letters*, 2023.

[19] J. Sun, Y. Jiang, J. Qiu, P. Nobel, M. J. Kochenderfer, and M. Schwager, “Conformal prediction for uncertainty-aware planning with diffusion dynamics model,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[20] J. Wang, J. Tong, K. Tan, Y. Vorobeychik, and Y. Kantaros, “Conformal temporal logic planning using large language models: Knowing when to do what and when to ask for help,” *arXiv preprint arXiv:2309.10092*, 2023.

[21] A. Z. Ren, A. Dixit, A. Bodrova, S. Singh, S. Tu, N. Brown, P. Xu, L. Takayama, F. Xia, J. Varley *et al.*, “Robots that ask for help: Uncertainty alignment for large language model planners,” *arXiv preprint arXiv:2307.01928*, 2023.

[22] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, “A survey of monte carlo tree search methods,” *IEEE Transactions on Computational Intelligence and AI in games*, vol. 4, no. 1, pp. 1–43, 2012.

[23] S. Junges, N. Jansen, and S. A. Seshia, “Enforcing almost-sure

reachability in POMDPs," in *International Conference on Computer Aided Verification*. Springer, 2021, pp. 602–625.

[24] J. Amirian, B. Zhang, F. V. Castro, J. J. Baldelomar, J.-B. Hayet, and J. Pettre, "Opentraj: Assessing prediction complexity in human trajectories datasets," in *Asian Conference on Computer Vision (ACCV)*, no. CONF. Springer, 2020.

APPENDIX

Lemma 1: The output of Algorithm 1, denoted by $\{W_t^\tau\}_{\tau=1}^H$, comprises a set of winning regions, with each W_t^τ representing a winning region for an $(H-\tau)$-step horizon.

Proof: We prove the lemma by induction on the increasing horizon of winning regions.

Base case: By the definition of W_t^H, for any belief state b_{t+H} whose belief support $supp(b_{t+H}) \in W_t^H$, every state $s_{t+H} \in supp(b_{t+H})$ satisfies the safety constraint, i.e., $c(s_{t+H}, \hat{X}_t^H) \geq L \cdot C_{t+H}^H$. Thus, W_t^H is a winning region for zero-step horizon.

Inductive step: Assume that W_t^τ is the winning region for an $(H-\tau)$-step horizon. By definition, for every belief state $b_{t+\tau}$ whose belief support $supp(b_{t+\tau}) \in W_t^\tau$, there exists an $(H-\tau)$-step horizon winning policy $\pi_{t+\tau}$ originating from $b_{t+\tau}$. For every belief state $b_{t+\tau-1}$ with belief support $supp(b_{t+\tau-1}) \in W_t^{\tau-1}$, there exists at least one action $a_{t+\tau-1} \in A$ that leads to a successor belief support state

$$supp(b_{t+\tau}) \in \mathrm{post}(\langle supp(b_{t+\tau-1}), \tau-1\rangle, a_{t+\tau-1}) \subseteq W_t^\tau.$$

We can augment policy $\pi_{t+\tau}$ into an $(H-\tau+1)$-step horizon winning policy $\pi_{t+\tau-1}$ with $\pi_{t+\tau-1}(b_{t+\tau-1}) = a_{t+\tau-1}$. Thus, $W_t^{\tau-1}$ is the winning region for an $(H-\tau+1)$-step horizon.

Conclusion: By induction, we have proved that Algorithm 1 outputs a set of winning regions $\{W_t^\tau\}_{\tau=1}^H$, with each W_t^τ having an $(H-\tau)$-step winning horizon. ∎

Theorem 1: Given a POMDP model $\mathcal{M}$ for a robotic agent with initial belief state b_0, the unknown random trajectories $X \sim \mathcal{D}$ of N dynamic agents with a prediction horizon H, and a failure probability $\delta \in (0,1)$, the policy π^* computed by Algorithm 2 achieves the maximal expected return $V^{\pi^*}(b_0)$ while ensuring safety, i.e., $\phi^{\pi^*}(b_0) \geq 1-\delta$.

Proof: Let $a_t = \pi^*(b_t)$ denote the action selected by the policy π^* at timestep t computed via Algorithm 2. By construction, a_t is a safe action leading solely to successor belief supports within the winning region W_t^1, which, according to Lemma 1, has an $(H-1)$-step winning horizon. Thus, for each state $s_{t+1} \in supp(b_{t+1})$, we have

$$c(s_{t+1}, \hat{X}_t^1) \geq L \cdot C_{t+1}^1. \tag{8}$$

Based on Equation 8 and thanks to Lipschitz continuity of function c, we derive that

$$\begin{aligned} 0 &\leq c(s_{t+1}, \hat{X}_t^1) - L \cdot C_{t+1}^1 &(9)\\ &\leq c(s_{t+1}, X_{t+1}) + L \cdot \|X_{t+1} - \hat{X}_t^1\| - L \cdot C_{t+1}^1. &(10) \end{aligned}$$

Hence, $\|X_{t+1} - \hat{X}_t^1\| \leq C_{t+1}^1$ is a sufficient condition for $c(s_{t+1}, X_{t+1}) \geq 0$, that is,

$$Pr(c(s_{t+1}, X_{t+1}) \geq 0) \mid \|X_{t+1} - \hat{X}_t^1\| \leq C_{t+1}^1) = 1. \tag{11}$$

Thanks to the law of total probability, we have

$$Pr(c(s_{t+1}, X_{t+1}) \geq 0) \geq Pr(\|X_{t+1} - \hat{X}_t^1\| \leq C_{t+1}^1). \tag{12}$$

With the assistance of Corollary 3 in [8], it can be shown that the ACP regions C_{t+1}^1 computed as per Section III-A guarantee that

$$\frac{1}{T}\sum_{t=0}^{T-1} Pr(\|X_{t+1} - \hat{X}_t^1\| \leq C_{t+1}^1) \geq 1 - \delta - p_1 \tag{13}$$

with constant $p_1 := \frac{\lambda_0^1 + \alpha}{T \cdot \alpha}$, where λ_0^1 is the constant initial value in Equation 6 and T is the number of times Equation 6 is applied. Combining Equation 12 and 13, we have

$$\frac{1}{T}\sum_{t=0}^{T-1} Pr(c(s_{t+1}, X_{t+1}) \geq 0) \geq 1 - \delta - p_1. \tag{14}$$

By definition of Equation 4, we have

$$\rho(b_{t+1}, X_{t+1}) = \sum_{s \in supp(b_{t+1})} b_{t+1}(s) \cdot \mathbb{I}_{\{c(s, X_{t+1}) \geq 0\}}. \tag{15}$$

Since Equation 14 holds for all $s \in supp(b_{t+1})$, we have

$$\begin{aligned} &\frac{1}{T}\sum_{t=0}^{T-1} \rho(b_t, X_t) \\ &= \frac{1}{T}\sum_{t=0}^{T-1} b_{t+1}(s) \cdot Pr(c(s_{t+1}, X_{t+1}) \geq 0) \\ &\geq 1 - \delta - p_1. \end{aligned} \tag{16}$$

Since $\lim_{T\to\infty} p_1 = 0$, we have

$$\phi^\pi(b_0) = \mathbb{E}_\pi[\lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \rho(b_t, X_t) \mid b_0, X_t \sim \mathcal{D}] \geq 1 - \delta.$$

Moreover, $a_t = \pi^*(b_t)$ is selected as the best action that achieves the maximum value of $\mathcal{V}(h_t a)$, which calculates the expected return from all simulations starting at h_t, among all safe actions enabled at timestep t. Thus, given a substantial number of simulations, the expected return $V^{\pi^*}(b_0)$ at the initial belief state b_0 is optimal.

In conclusion, we have proved that the policy π^* computed by Algorithm 2 achieves the maximal expected return $V^{\pi^*}(b_0)$ while ensuring safety, i.e., $\phi^{\pi^*}(b_0) \geq 1 - \delta$. ∎