Safe Planning in Unknown Environments Using Conformalized Semantic Maps

David Smith Sundarsingh, Yifei Li, *Graduate Student Member, IEEE*, Tianji Tang, George J. Pappas, *Fellow, IEEE*, Nikolay Atanasov, *Senior Member, IEEE*, and Yiannis Kantaros, *Member, IEEE*

***Abstract*—This paper addresses semantic planning problems in unknown environments under perceptual uncertainty. The environment contains multiple unknown semantically labeled regions or objects, and the robot must reach desired locations while maintaining class-dependent distances from them. We aim to compute robot paths that complete such semantic reach-avoid tasks with user-defined probability despite uncertain perception. Existing planning algorithms either ignore perceptual uncertainty—thus lacking correctness guarantees—or assume known sensor models and noise characteristics. In contrast, we present the first planner for semantic reach-avoid tasks that achieves user-specified mission completion rates without requiring any knowledge of sensor models or noise. This is enabled by quantifying uncertainty in semantic maps—constructed on-the-fly from perceptual measurements—using conformal prediction in a model- and distribution-free manner. We validate our approach and the theoretical mission completion rates through extensive experiments, showing that it consistently outperforms baselines in mission success rates.**

***Index Terms*—Planning under uncertainty, reactive and sensor-based planning, AI-based methods.**

I. Introduction

EXTENSIVE research on autonomous robot navigation has resulted in numerous planners capable of designing trajectories that reach known goal regions while avoiding obstacles [1], [2]. Recent advances in AI-enabled robot vision and semantic mapping have created new opportunities to move beyond these well-studied geometric planning approaches toward semantic planning, where task completion requires reasoning not only about the geometric structure of the environment (e.g., obstacle locations) but also its semantic structure (e.g., object categories or terrain types) [3], [4].

This paper focuses on semantic planning problems in environments populated with multiple objects and regions whose locations and semantic categories are initially unknown. The robot's goal is to reach desired destinations while maintaining category-dependent safety distances from these semantic regions and objects. Recent planning algorithms tackle such tasks by leveraging AI-enabled perception systems (e.g., object recognition algorithms) to collect geometric and semantic observations [5], [6], [7], [8], [9], [10]. These approaches typically either (i) employ deep learning methods to directly map perceptual inputs to robot actions (e.g., [8], [10], [11]) or (ii) leverage semantic SLAM techniques [12], [13], [14] to build semantic maps that are used by downstream planners (e.g., [5], [6], [15]). Despite impressive empirical performance, in these architectures the planning layer generally treats outputs from the perception or mapping modules as ground truth, ignoring their inherent uncertainty. As a result, these approaches lack mission completion guarantees. Attempts to model perceptual uncertainty within semantic and geometric planning have been investigated in [5], [6], [16] and [17], [18], [19], [20], respectively. However, these methods often depend on unreliable proxies such as SLAM-derived confidence scores that do not represent true probabilities [21], or assume known sensor models and Gaussian noise—assumptions that may not hold in practice.

This paper addresses these limitations by developing a semantic planner for unknown environments that completes semantic reach-avoid tasks with a user-defined probability under perceptual uncertainty, while remaining agnostic to sensor models and noise characteristics. Robots are equipped with imperfect perception systems (e.g., cameras with semantic segmentation) that provide geometric and semantic observations. Our framework consists of three main components; see Fig. 1. First, we leverage existing semantic mapping algorithms that capture dense semantic and geometric environmental information and generate a multi-class semantic map of the environment [13]. Second, we quantify uncertainty of the generated semantic maps using Conformal Prediction (CP), a statistical, model- and distribution-free tool [22]. This produces a set of maps guaranteed to contain the ground truth with a user-defined probability. Third, we develop a semantic planner that balances exploitation and exploration. In exploitation mode, the planner selects the most conservative map from the CP set and generates paths using any search- or sampling-based planner [1], [2]. When uncertainty is too high for a feasible solution to exist, the planner switches to exploration mode to reduce uncertainty (modeled by the size of the CP set) until a feasible path can be found. We show

Received 29 September 2025; accepted 3 February 2026. Date of publication 27 February 2026; date of current version 9 March 2026. This article was recommended for publication by Associate Editor Q. Lin and Editor L. Pallottino upon evaluation of the reviewers' comments. This work was supported in part by the ARL under Grant DCIST CRA W911NF-17-2-0181 and in part by the NSF CAREER under Award CNS 2340417. *(Corresponding author: David Smith Sundarsingh.)*

David Smith Sundarsingh, Yifei Li, and Yiannis Kantaros are with the Department of Electrical and Systems Engineering, Washington University in St. Louis, St. Louis MO 63108 USA (e-mail: d.s.sundarsingh@wustl.edu; liyifei@wustl.edu; ioannisk@wustl.edu).

Tianji Tang and Nikolay Atanasov are with the Department of Electrical and Computer Engineering, University of California, San Diego CA 92093 USA (e-mail: tit006@ucsd.edu; natanasov@ucsd.edu).

George J. Pappas is with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia PA 19104 USA (e-mail: pappasg@seas.upenn.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2026.3668466, provided by the authors.

Digital Object Identifier 10.1109/LRA.2026.3668466

2377-3766 © 2026 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:01:10 UTC from IEEE Xplore. Restrictions apply.

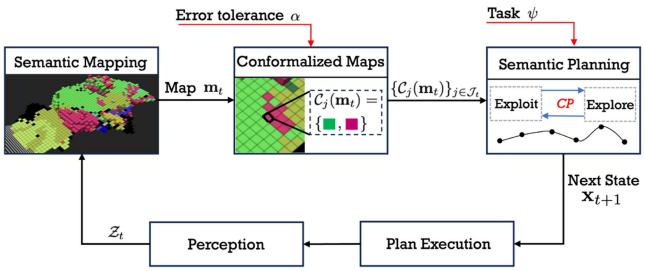


Fig. 1. Illustration of the proposed planning framework for semantic reach-avoid tasks ψ. Given categorical and range measurements, a semantic map is constructed [13]. To account for map uncertainty caused by perceptual imperfections, conformal prediction is employed to generate sets of maps that contain the ground-truth environment with a user-defined probability. These sets are then used by a planner to generate uncertainty-aware paths that complete the assigned tasks with a user-defined probability $1-\alpha$.

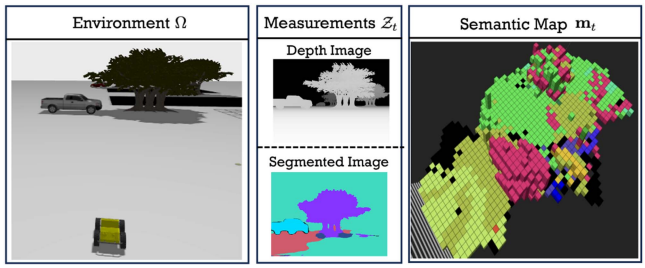


Fig. 2. The mapping algorithm uses depth and categorical measurements $\mathcal{Z}_t$, provided as a depth image and a segmented image, respectively, to generate a 3D semantic grid map $\mathbf{m}_t$. Each grid cell is associated with a semantic category, and its color corresponds to that category.

both theoretically and empirically that our method achieves desired mission completion rates. We also conduct experiments highlighting the exploitation–exploration trade-offs as the required success probability increases. Our comparative experiments also demonstrate that our approach outperforms baselines that either treat mapping outputs as ground truth or rely on uncertainty metrics provided by the mapping algorithm.

Related Work: (i) Perception-based Planning: Numerous planners for unknown environments have been proposed. However, these either neglect perceptual/mapping uncertainty and, thus, lack correctness guarantees [7], [8], [9], [10], [11], [23], or assume known sensor models and noise characteristics to design probabilistically correct paths [5], [6], [17], [18], [19], [20]. In contrast, our planner achieves user-specified mission completion rates without assuming any knowledge of sensor models or noise. The closest work to ours is [24], which employs CP to quantify the uncertainty of object detectors and design probabilistically correct paths. However, [24] focuses solely on geometric planning and cannot straightforwardly handle combined geometric and semantic uncertainty. *(ii) CP for Verified Autonomy:* CP has been applied to quantify uncertainty in various AI-enabled components (e.g., perception, motion prediction, and LLMs) [25], [26], [27], [28], [29], [30], [31]. Inspired by these works, we propose the first use of CP to quantify semantic map uncertainty enabling probabilistically correct semantic planning.

Contributions: First, we introduce the first semantic planner for unknown environments that achieves user-defined mission completion rates under perceptual uncertainty, while remaining agnostic to sensor models and noise characteristics. Second, we present the first application of CP to quantify the uncertainty of semantic maps generated by existing mapping algorithms, also in a model-agnostic manner. Third, we present experiments that validate the theoretical mission completion guarantees and show that our method consistently outperforms baselines in terms of mission success rates.

II. Problem Statement

A. Robot & Environment Modeling

Consider a robot with configuration $\mathbf{x}_t \in \mathcal{X}$ at discrete time $t \geq 0$. We assume that the configuration $\mathbf{x}_t$ is known (e.g., from encoders and localization). The robot operates in an environment $\Omega \subseteq \mathbb{R}^d$, $d \in \{2,3\}$, which is represented as a collection of disjoint sets $\mathcal{L}_k \subseteq \Omega$, each associated with a semantic category $k \in \mathcal{K} = \{0,1,\ldots,K\}$ (e.g., cars, walls, trees). Here, $\mathcal{L}_0$ denotes free space, and $\mathcal{L}_k$ are fixed over time and unknown to the robot.

B. Reachability Task With Semantic Safety Constraints

The robot is tasked with reaching a known set of goal states $\mathcal{X}_g \subseteq \mathcal{X}$ while maintaining class-dependent safe distances d_k from regions $\mathcal{L}_k$. We denote the assigned task by ψ. As an example, consider a safety specification requiring the robot to maintain a distance of 1 m from $\mathcal{L}_{\text{tree}}$ and 3 m from $\mathcal{L}_{\text{car}}$ while navigating towards a desired destination. Such semantic reach-avoid tasks can be completed by following a finite-horizon path $\mathbf{x}_{0:H}$, defined as a sequence of robot states, i.e., $\mathbf{x}_{0:H} = \mathbf{x}_0,\ldots,\mathbf{x}_t,\ldots,\mathbf{x}_H$, for some horizon H, where (i) $\mathbf{x}_H \in \mathcal{X}_g$, and (ii) all $\mathbf{x}_t$ satisfy the safety constraints, i.e.,

$$\min_{\text{ı}\in\mathcal{L}_k} \|\mathbf{x}_t|_\Omega - \text{ı}\|_2 \geq d_k, \quad \forall t \in \{0,\ldots,H\}, \tag{1}$$

for all $\mathcal{L}_k \subseteq \Omega$, where $k \in \mathcal{K}\setminus\{0\}$. In (1), $\mathbf{x}_t|_\Omega$ denotes the robot's position, i.e., the projection of the robot state onto Ω. Completion of ψ using a robot path $\mathbf{x}_{0:H}$ is denoted by $\mathbf{x}_{0:H} \models \psi$, where $\models$ denotes the satisfaction relation.

C. Perception System and Semantic Mapping

The robot is equipped with a sensor of range $r_{\max} > 0$ that provides information about the distances to, and semantic categories of, surrounding objects along a set of rays. We denote by $\mathcal{Z}_t := \{z_{t,b}\}_b$ the set of measurements collected along ray b at time t, where each measurement is given by $z_{t,b} = \{r_{t,b}, y_{t,b}\}$, with $r_{t,b} \in \mathbb{R}_{\geq 0}$ and $y_{t,b} \in \mathcal{K}$ denoting the range and categorical measurements, respectively.

These labeled range measurements, obtained by processing RGB-D images, can be fused online to build a semantic representation of Ω using semantic mapping algorithms. We adopt [13], which models the environment as a multi-class grid map $\mathbf{m}$ defined as a collection of cells $j \in \mathcal{J} = \{1,\ldots,M\}$, each labeled with a category $m_j \in \mathcal{K}$ and represented by its center $c_j \in \Omega$; see Fig. 2. To account for perceptual noise, each cell m_j maintains a probability mass function (PMF) $p_t(m_j) = p(m_j|\mathbf{x}_{0:t}, \mathcal{Z}_{0:t})$ over $\mathcal{K}$, conditioned on the measurements $\mathcal{Z}_{0:t}$ collected along the robot path $\mathbf{x}_{0:t}$. If at some t, no measurement corresponds to cell j, $p_t(m_j) = p_{t-1}(m_j)$. Accordingly, the distribution of map is defined as $p_t(\mathbf{m}) = \prod_{j\in\mathcal{J}} p_t(m_j)$. For brevity, we

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:01:10 UTC from IEEE Xplore. Restrictions apply.

denote by σ a mapping algorithm that takes as input the distributions $p_{t-1}(m_j)$ and the new measurement $\mathcal{Z}_t$ collected at $\mathbf{x}_t$, and outputs the updated distributions $p_t(m_j), \forall j \in \mathcal{J}_t$, i.e., $p_t(\mathbf{m}) = \sigma(\Omega, p_{t-1}(\mathbf{m}), \mathcal{Z}_t)$, where $\mathcal{J}_t \subseteq \mathcal{J}$ is the set of indices of the grid cells that have been detected till time t; map update at $t = 0$ occurs using a prior $p_{-1}(\mathbf{m})$. Hereafter, when it is clear from the context, we refer to the random variable $\mathbf{m}_t$ without explicitly specifying its density $p_t(\mathbf{m})$.

D. Problem Statement

Consider any user-defined initial robot state $\mathbf{x}_0$ and semantic reach-avoid task ψ for an environment Ω. We assume that Ω is sampled from an unknown distribution $\mathcal{D}$, from which a finite set of i.i.d environments can be sampled. The robot does not know Ω; instead it has access to a prior map $\mathbf{m}_{-1}$ that can be updated online using the mapping algorithm σ. We aim to design a planning algorithm that, given an environment sampled from $\mathcal{D}$, can generate a robot path $\mathbf{x}_{0:H}$, for some $H > 0$, that completes ψ with a user-specified probability $1 - \alpha$, given the sequence of maps $\mathbf{m}_{0:H}$ updated online. Formally, we address the following problem:

Problem II.1: Consider an unknown environment $\Omega \sim \mathcal{D}$ with prior information $\mathbf{m}_{-1}$, and any initial robot state $\mathbf{x}_0$ and semantic reach-avoid task ψ defined over Ω. Given a mapping algorithm σ, design a planning algorithm that generates a robot path $\mathbf{x}_{0:H}$, for some $H > 0$, ensuring that

$$\mathbb{P}_{\Omega\sim\mathcal{D}}(\mathbf{x}_{0:H} \models \psi \,|\, \mathbf{m}_{0:H}) \geq 1 - \alpha, \tag{2}$$

where $1 - \alpha \in (0, 1)$ is a user-specified success rate.

We note that the probabilistic task completion statement in (2) accounts for errors at the mapping level, but not necessarily at the sensor level. For instance, certain sensor rays b may fail to return a measurement from an object occupying cell j (e.g., due to malfunctions). In such cases, the PMF of m_j is never updated, creating the false impression that it lies in free space. The guarantee in (2) does not cover these sensor-level errors; rather, it addresses labeling and mapping errors in cells m_j for which at least one associated measurement has been collected. Failures can also occur due to objects that (i) are close enough to the robot, leading to violations of (1) but they (ii) lie outside the sensor's field of view. To exclude such sensor-based errors, we make the following assumption.

Assumption II.2 (Sensors): (i) The robot is equipped with an *omnidirectional* sensor of range $r_{\max}$ generating $\mathcal{Z}_t$. (ii) The sensor range $r_{\max}$ satisfies $r_{\max} \geq d_k$ for all $k \in \mathcal{K}$ (iii) Any cell j in the map within the sensor's field of view always has a measurement associated with it. Due to (iii), if no measurements are obtained for cell j (lying within the sensor range), then it corresponds to free-space.

III. Safe Semantic Planning in Unknown Environments

A natural approach to Problem II.1 is to plan directly over the distribution of the map $\mathbf{m}_t$, as in [5], [6]. However, in practice, the sensor models used to update the map distribution are inaccurate, so the PMFs $p_t(m_j)$ do not represent true probabilities. As a result, planning directly over $\mathbf{m}_t$ may compromise the probabilistic correctness of the generated paths. To overcome this limitation, we propose a new planner that makes no assumptions about the sensor model and noise characteristics. In Section III-A, we employ conformal prediction (CP) to 'calibrate' $\mathbf{m}_t$ by computing prediction sets for each cell $j \in \mathcal{J}_t$ that contain the true label with user-defined confidence. These sets are then used in Section III-B to design paths $\mathbf{x}_{0:H}$ that solve Problem II.1.

A. Conformalizing Semantic Maps

As discussed earlier, the distribution of $\mathbf{m}_t$ does not represent true probabilities, compromising the correctness of paths designed only using $\mathbf{m}_t$. To address this, for any test environment $\Omega_{\text{test}} \sim \mathcal{D}$, we aim to design sets $\mathcal{C}_j(\mathbf{m}_t) \subseteq \mathcal{K}$ of labels for each cell $j \in \mathcal{J}_t$ detected along a path $\mathbf{x}_{0:H}$ and time step $t \in \{0, \dots, H\}$ satisfying the following property:

$$\mathbb{P}_{\Omega\sim\mathcal{D}}\left(C_{0:H} \mid \mathbf{m}_{0:H}\right) \geq 1 - \alpha, \tag{3}$$

where $C_{0:H} = \bigwedge_{\forall j\in\mathcal{J}_t, t\in[0:H]}(k_j^{\text{gt}} \in \mathcal{C}_j(\mathbf{m}_t))$ and $k_j^{\text{gt}} \in \mathcal{K}$ is the true label of the region occupying cell j. In words, (3) guarantees the inclusion of the true labels k_j^{gt} in the sets $\mathcal{C}_j(\mathbf{m}_t), \forall j \in \mathcal{J}_t$ in $\mathbf{m}_t$, and $\forall t \in \{0, \dots, H\}$, and any robot path $\mathbf{x}_{0:H}$ followed to construct $\mathbf{m}_{0:H}$.

To construct these sets for Ω_{test}, we employ CP to quantify uncertainty in the outputs of the mapping algorithm σ. Applying CP requires (i) a set of calibration environments $\{\Omega_i\}_{i=1}^D \sim \mathcal{D}$ with known ground-truth labels k_j^{gt} for all $j \in \mathcal{J}$, and (ii) a non-conformity score (NCS) that captures the error of σ, which generates maps $\mathbf{m}_{0:H}$ along *any* path $\mathbf{x}_{0:H}$. It is crucial that the NCS accounts for mapping errors along arbitrary robot paths, rather than a single fixed path, since otherwise (3) would hold only for that path. Thus, we adopt the following NCS, which is path (and, consequently, task)-agnostic and captures the worst-case mapping error, associated with the calibration environment Ω_i:

$$s_i = \max_{\mathbf{x}_{0:H}\in\mathcal{P}; j\in\mathcal{J}_t; t\in[0,H]} \left(1 - p_t(m_j = k_j^{\text{gt}})\right), \tag{4}$$

where $\mathcal{P}$ is the set of all possible finite paths in Ω_i. In (4), the term $p_t(m_j = k_j^{\text{gt}})$ denotes the probability that cell j has the ground-truth label according to the map $\mathbf{m}_t$. Consequently, $1 - p_t(m_j = k_j^{\text{gt}})$ models the mapping error at cell j after collecting measurements along a specific path $\mathbf{x}_{0:t}$. The maximum operator in (4) computes the worst-case error, i.e., the largest error across all cells $j \in \mathcal{J}_t$, all time steps $t \in [0, H]$, and all paths $\mathbf{x}_{0:H} \in \mathcal{P}$.[1] We compute the NCSs $\{s_i\}_{i=1}^D$ for all calibration environments and determine their $\frac{\lceil (D+1)(1-\alpha)\rceil}{D}$-quantile, denoted by $\hat{s}$.

For an unseen test environment $\Omega_{\text{test}} \sim \mathcal{D}$ with unknown true labels for cells j, at each step t we compute the following prediction set of labels $\forall j \in \mathcal{J}_t$ (see Algorithm 1):

$$\mathcal{C}_j(\mathbf{m}_t) = \{k \in \mathcal{K} | p_t(m_j = k) \geq 1 - \hat{s}\}, \tag{5}$$

collecting all labels $k \in \mathcal{K}$ satisfying $p_t(m_j = k) \geq 1 - \hat{s}$ where p_t refers to the PMF computed by σ [line 3, Algorithm 1]. In Section IV, we show that these prediction sets satisfy (3).

[1] Computing (4) exactly requires evaluating the maximum over an infinite number of paths, which is impractical. As in [24], in practice the NCS is approximated using a finite set of paths $\mathcal{P}$; see Section V-A for details.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:01:10 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1: Conformalization of Semantic Map.

1: **Input**: Map m_t, score quantile $\hat{s}$, semantic classes $\mathcal{K}$
2: **for** $j \in \mathcal{J}_t$ **do**
3: $\quad \mathcal{C}_j(\mathbf{m}_t) = \{k \in \mathcal{K} | p_t(m_j = k) \geq 1 - \hat{s}\}$
return $\{\mathcal{C}_j(m_t)\}_{j \in \mathcal{J}_t}$

B. Planning Over Conformalized Semantic Maps

Consider an environment $\Omega \sim \mathcal{D}$. Our goal is to compute a plan $\mathbf{x}_{0:H}$ satisfying any user-defined mission specification ψ with probability at least $1 - \alpha$. This path will be constructed incrementally as the map $\mathbf{m}_t$ is updated online; see Algorithm 2. First, we collect the calibration dataset $\{\Omega_i\}_{i=1}^D$, and compute $\hat{s}$ as described in Section III-A [lines 3-4, Algorithm 2]. Suppose at time $t > 0$, the robot is at state $\mathbf{x}_t$, collects measurements $\mathcal{Z}_t$, and constructs the map $\mathbf{m}_t$ [lines 6-7, Algorithm 2]. Using $\mathbf{m}_t$, the prediction sets $\mathcal{C}_j(\mathbf{m}_t)$ are constructed for all cells $j \in \mathcal{J}_t$ [line 8, Algorithm 2]. We now discuss how the next robot state $\mathbf{x}_{t+1}$ is computed using these sets [lines 9–14, Algorithm 2]. The next state $\mathbf{x}_{t+1}$ must be chosen so that it satisfies (1). Since the regions $\mathcal{L}_k$ are unknown, satisfaction of (1) must be checked using the map $\mathbf{m}_t$. Specifically, given $\mathbf{m}_t$, (1) can be re-written as:

$$\|\mathbf{x}_{t+1}|_\Omega - c_j\| \geq d_{k_j} + r_m, \tag{6}$$

for all cells $j \in \mathcal{J}_t$, where r_m is the grid resolution, $c_j \in \Omega$ is the center of cell j and k_j is the label assigned to cell j.[2] Note that cells $j \in \mathcal{J} \setminus \mathcal{J}_t$ within the sensor range $r_{\max}$ are free space due to Assumption II.2. The next state $\mathbf{x}_{t+1}$ can be chosen using any existing planner if the true label k_j^{gt} of each cell j were known to the robot, which is not the case here. To address this, we leverage sets $\mathcal{C}_j(\mathbf{m}_t)$, constructed as shown in (5), that contain the true label with user-defined confidence. From each set, we select the most conservative label, i.e., $k_j = \arg\max_{k \in \mathcal{C}_j(\mathbf{m}_t)} d_k$, for every cell $j \in \mathcal{J}_t$. All cells $j \in \mathcal{J} \setminus \mathcal{J}_t$ outside the sensor range are considered part of the free space i.e., $k_j = 0$, for all $j \in \mathcal{J} \setminus \mathcal{J}_t$. Given this assignment of labels, the planner adaptively switches between exploitation and exploration modes as follows.

Exploitation Phase: Using an existing motion planner [1], [2], [32], we first check whether there exists a robot path $\mathbf{p} = \mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_T$, with $\mathbf{p}_n \in \mathcal{X}$ for all $n \in \{0, \ldots, T\}$ and some $T \geq 0$ that completes ψ under the assigned labels k_j [line 9, Algorithm 2]. The path must satisfy $\mathbf{p}_0 = \mathbf{x}_t$, $\mathbf{p}_T \in \mathcal{X}_g$, and $\|\mathbf{p}_n|_\Omega - c_j\| \geq d_{k_j} + r_m, \forall n$. If such a feasible plan exists, the robot selects $\mathbf{x}_{t+1} = \mathbf{p}_1$, moves towards it, updates the time step to $t + 1$, and repeats the process [lines 12–14, Algorithm 2].

Exploration Phase: If no feasible solution is found during the exploitation phase, the planner switches to exploration [line 10, Algorithm 2]. In this mode, any exploration algorithm can be employed to generate paths $\mathbf{p}$ that optimize information-theoretic objectives (e.g., mutual information) subject to the constraints in (6); see, e.g., [13], [33]. Given such a path $\mathbf{p}$, the planner selects $\mathbf{x}_{t+1} = \mathbf{p}_1$, sends it to the robot, updates the time step to $t + 1$, and repeats the process [lines 9–14, Algorithm 2]. The goal is for exploration paths to reduce uncertainty (i.e., prediction set sizes) for the robot to re-enter exploitation mode.

[2] The grid resolution is added to ensure that the robot never gets close to an object even if the object occupies the entirety of the grid cell.

Algorithm 2: Safe Planning Over Conformalized Maps.

1: **Input**: Prior map $\mathbf{m}_{-1}$; Task ψ; Initial state $\mathbf{x}_0$; Set of labels $\mathcal{K}$
2: Initialize $t = 0$
3: Collect calibration environments $\{\Omega_i\}_{i=1}^D$
4: Compute NCSs $\{s_i\}_{i=1}^D$ and quantile $\hat{s}$
5: **while** $\mathbf{x}_t \notin \mathcal{X}_g$ **do**
6: $\quad \mathcal{Z}_t = \texttt{Measure}(\Omega, \mathbf{x}_t)$
7: $\quad \mathbf{m}_t = \sigma(\Omega, \mathbf{m}_{t-1}, \mathcal{Z}_t)$
8: $\quad \{\mathcal{C}_j(\mathbf{m}_t)\}_{j \in \mathcal{J}_t} = \texttt{Conformalize}(\mathbf{m}_t, \hat{s}, \mathcal{K})$ (see Algorithm 1)
9: $\quad \mathbf{p} = \texttt{Plan}(\mathbf{x}_t, \mathcal{X}_g, \{\mathcal{C}_j(\mathbf{m}_t)\}_{j \in \mathcal{J}_t})$
10: $\quad$ **if** $\mathbf{p} = \emptyset$ **then**
11: $\quad\quad \mathbf{p} = \texttt{Explore}(\mathbf{x}_t, \{\mathcal{C}_j(\mathbf{m}_t)\}_{j \in \mathcal{J}_t})$
12: $\quad \mathbf{x}_{t+1} = \mathbf{p}(1)$
13: $\quad \texttt{Execute}(\mathbf{x}_{t+1})$
14: $\quad t = t + 1$

During both exploration and exploitation we also ensure that $\|\mathbf{x}_{t+1} - \mathbf{x}_t\| < r_{\max}$ to eliminate cases where the next state may lie in the unexplored part of the map. The above process terminates once $\mathbf{x}_{t+1} \in \mathcal{X}_g$, and the robot reaches $\mathbf{x}_{t+1}$. Concatenation of these states across time yields $\mathbf{x}_{0:H}$.

IV. Performance Guarantees

In this section, we show that our planner addresses Problem II.1, i.e., the path $\mathbf{x}_{0:H}$ generated by Algorithm 2 completes ψ with probability $1 - \alpha$. This result builds upon the fact that the sets $\mathcal{C}_j(\mathbf{m}_t)$, constructed by Algorithm 1, satisfy (3), i.e., they jointly contain the true labels for all $j \in \mathcal{J}_t$, $t \in [0, H]$ and any path $\mathbf{x}_{0:H}$ with probability at least $1 - \alpha$.

Proposition IV.1 (Mapping Guarantees): Consider a test environment $\Omega_{\text{test}} \sim \mathcal{D}$ with unknown true labels $\{k_j^{\text{gt}}\}_{j \in \mathcal{J}}$. Given any path $\mathbf{x}_{0:H}$ in Ω_{test} yielding a sequence of maps $\mathbf{m}_{0:H}$, the prediction sets $\{\mathcal{C}_j(\mathbf{m}_t)\}_{j \in \mathcal{J}_t}$, computed at every $t \in \{0, \ldots, H\}$ constructed by Algorithm 1 satisfy (3), for a user-defined $\alpha \in (0, 1)$.

Proof: Using a standard CP argument [24], [34], we have:

$$\mathbb{P}_{\Omega_{\text{test}} \sim \mathcal{D}}\left(s_{\text{test}} \leq \hat{s} | \, \mathbf{m}_{0:H}\right) \geq 1 - \alpha, \tag{7}$$

where s_{test} is the NCS associated with Ω_{test}. Note that s_{test} is unknown as the ground truth labels k_j^{gt} for cells j are unknown. By definition of the NCS in (4), we have that $s_{\text{test}} = \max_{\mathbf{x}_{0:H} \in \mathcal{P}, j \in \mathcal{J}_t, t \in [0:H]} 1 - p_t(m_j = k_j^{gt})$. Plugging this into (7) yields the following result:

$$\mathbb{P}_{\Omega_{\text{test}} \sim \mathcal{D}}\left(\max_{\mathbf{x}_{0:H} \in \mathcal{P}, j \in \mathcal{J}_t, t \in [0:H]} (1 - p_t(m_j = k_j^{gt}) \leq \hat{s} \mid \mathbf{m}_{0:H})\right) \geq 1 - \alpha.$$

Thus, for any path $\mathbf{x}_{0:H} \in \mathcal{P}$, we have that

$$\mathbb{P}_{\Omega_{\text{test}} \sim \mathcal{D}}\left(\bigwedge_{\forall j \in \mathcal{J}_t, \forall t \in [0:H],} p_t(m_j = k_j^{gt}) \geq 1 - \hat{s} \mid \mathbf{m}_{0:H}\right) \geq 1 - \alpha. \tag{8}$$

Recall from (5) that the prediction sets $\mathcal{C}_j(\mathbf{m}_t)$ collect all labels $k \in \mathcal{K}$ satisfying $p_t(m_j = k) \geq 1 - \hat{s}$. Combining this

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:01:10 UTC from IEEE Xplore. Restrictions apply.

with (8) results in (3), i.e., $\mathbb{P}_{\Omega_{\text{test}}\sim\mathcal{D}}(C_{0:H}|\mathbf{m}_{0:H}) \geq 1-\alpha$, where $C_{0:H} = \bigwedge_{\forall j\in\mathcal{J}_t, t\in[0:H]}(k_j^{\text{gt}} \in \mathcal{C}_j(\mathbf{m}_t))$.

Observe that the result in Proposition IV.1 does not rely on Assumption II.2, as it focuses only on map cells $j \in \mathcal{J}_t$ for which at least one measurement has been taken. Building on Proposition IV.1, we show that the path $\mathbf{x}_{0:H}$ generated by Algorithm 2 addresses Problem II.1. To show this result, we make the following assumptions about the exploration and planning algorithms employed in Algorithm 2.

Assumption IV.2: (i) The exploration algorithm is invoked a finite number of times before a path $\mathbf{x}_{0:H}$ is computed. Each invocation reduces mapping uncertaity-and consequently prediction set size-so that the planner switches back to exploitation mode. (ii) The planning algorithm used in the exploitation phase is complete, i.e., if a feasible path $\mathbf{p}$ exists that completes ψ under the assigned labels described in Section III-B, then the planner is guaranteed to find it. These assumptions ensure that mission failures can only be attributed to mapping errors.

Theorem IV.3 (Planning Guarantees): Consider a test environment $\Omega_{\text{test}} \sim \mathcal{D}$ with unknown true labels $\{k_j^{\text{gt}}\}_{j\in\mathcal{J}}$, and a mission specification ψ_{test}. Under Assumptions II.2 and IV.2, Algorithm 2 will generate a path $\mathbf{x}_{0:H}$ satisfying (2).

Proof: First, by Assumption IV.2, in any environment $\Omega_{\text{test}} \sim \mathcal{D}$, and for any task ψ_{test}, Algorithm 2 is guaranteed to terminate and return a path $\mathbf{x}_{0:H}$. Consequently, any failure to complete ψ_{test} can be attributed only to mapping errors. Second, consider the path $\mathbf{x}_{0:H}$ returned by Algorithm 2 for the task ψ_{test}, when operating in $\Omega_{\text{test}} \sim \mathcal{D}$. We show that $\mathbf{x}_{0:H}$ satisfies (2) by leveraging Proposition IV.1. Specifically, by construction of the exploration and exploitation modes in Algorithm 2, Assumption IV.2-(ii), and Assumption II.2 we have that: (i) all states $\mathbf{x}_t$ in $\mathbf{x}_{0:H}$ satisfy the imposed safety constraints given the labels $k_j = \arg\max_{k\in\mathcal{C}_j(\mathbf{m}_t)} d_k$ assigned to cells j, i.e., $\max_{k\in\mathcal{C}_j(\mathbf{m}_t)} d_k + r_m \leq \|\mathbf{x}_{t+1}|_\Omega - c_j\|, \forall j \in \mathcal{J}_t$ and (ii) $\mathbf{x}_H \in \mathcal{X}_g$. Now, let us assume that the ground truth label is included in the prediction sets for all $t \in \{0,\dots,H\}$. This equivalently means that the probability that $\mathbf{x}_{0:H}$ satisfies ψ with respect to the assigned labels, and implicitly the ground truth labels, is 1. The implicit satisfaction with respect to the ground truth is due to the consideration of the worst-case label in the prediction set that also contains the ground truth. We express this as follows:

$$\mathbb{P}_{\Omega_{\text{test}}\sim\mathcal{D}}\left(\mathbf{x}_{0:H} \models \psi | C_{0:H}, \mathbf{m}_{0:H}\right) = 1, \tag{9}$$

where $C_{0:H} = \bigwedge_{\forall j\in\mathcal{J}_t, t\in[0:H]}(k_j^{gt} \in \mathcal{C}_j(\mathbf{m}_t))$. Recall from Proposition IV.1, that $\mathbb{P}_{\Omega_{\text{test}}\sim\mathcal{D}}(C_{0:H}|\mathbf{m}_{0:H}) \geq 1-\alpha$. Combining this with (9) we get that $\mathbf{x}_{0:H}$ satisfies (2).

Remark IV.4 (Dataset-conditional guarantees): The guarantees in Proposition IV.1 and Theorem IV.3 are marginal, where the probabilities are over the randomness of the calibration set and the test environment. A dataset-conditional guarantee which holds for a fixed calibration set across environments can also be applied [35]. In such cases, Proposition IV.1 is modified such that with a probability of at least $1-\delta$ over the sampling of the calibration set$\{\Omega_i\}_{i=1}^D \sim \mathcal{D}$, $\mathbb{P}_{\Omega_{\text{test}}\sim\mathcal{D}}(\mathcal{C}_{0:H}|\mathbf{m}_{0:H}, \{\Omega_i\}_{i=1}^D) \geq Beta^{-1}_{D+1-v,v}(\delta)$, where $Beta^{-1}_{D+1-v,v}(\delta)$ is the δ-quantile of the Beta distribution with parameters, D - the size of the calibration dataset - and $v = \lfloor (D+1)\hat{\alpha} \rfloor$ and $\hat{\alpha}$ is selected so that $Beta^{-1}_{D+1-v,v}(\delta) \approx 1-\alpha$.

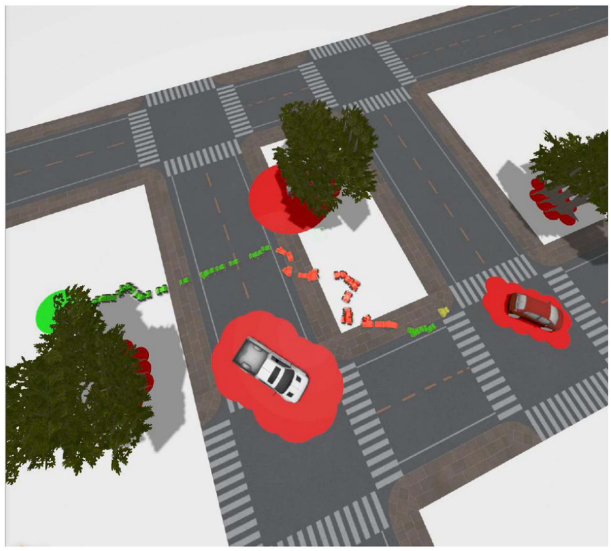


Fig. 3. Illustration of a path $\mathbf{x}_{0:H}$ to the goal region (green disk) consisting of an exploration sub-path (red) and two exploitation sub-paths (green). The red regions model the unsafe regions as per (9).

Similarly, Theorem IV.3 is modified, where with a probability of at least $1-\delta$ over the sampling of the calibration dataset, $\mathbb{P}_{\Omega_{\text{test}}\sim\mathcal{D}}(\mathbf{x}_{0:H} \models \psi|\mathbf{m}_{0:H}, \{\Omega_i\}_{i=1}^D) \geq Beta^{-1}_{D+1-v,v}(\delta)$.

V. Experimental Validation

In this section, we empirically validate the proposed approach. Sections V-A introduces the setup of our experiments. The empirical performance of our method is presented in Sections V-B–V-C, while Section V-D provides comparative results against uncertainty-informed and uncertainty-agnostic planners. Finally, Sections V-E–V-F evaluate our algorithm in out-of-distribution (OOD) real-world and simulation scenarios, highlighting potential failure cases. Demonstrations in simulation and real-world are provided in [36].

A. Setting up Experiments and the Proposed Planner

Robot: Our experiments are conducted using ROS 2 Humble [37] and the Gazebo (Ignition) simulator. The simulated platform is a Clearpath Husky skid-steer ground vehicle with a maximum linear speed of 1.0 m/s and a maximum yaw rate of 0.5 rad/s. It is equipped with a forward-facing RGB-D camera with a resolution of 640×480, a horizontal FoV of $90°$, a vertical FoV of $73.7°$, and a sensing range of $r_{\max} = 10$ m. Note that this sensor is not omnidirectional, as required in Assumption II.2, due to its FoV; consequently, our algorithm is tested in a more challenging setting.

Distribution $\mathcal{D}$: The distribution $\mathcal{D}$ is designed to generate environments as follows. All environments Ω have dimensions of 70×25 m containing objects from five semantic classes $\mathcal{K} = \{\texttt{person}, \texttt{car}, \texttt{truck}, \texttt{tree}, \texttt{free}\}$. Each environment is populated with one car, one truck, one human placed at randomized positions and orientations, with the human biased toward being placed near trees, and three parallel rows of three trees in fixed locations. The class-based safety constraints are: $d_{\text{person}} = 4$ m, $d_{\text{car}} = 1$ m, $d_{\text{truck}} = 2$ m, and $d_{\text{tree}} = 0.5$ m. This choice enables testing our algorithm in cases of partial occlusions. An example environment is shown in Fig. 3.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:01:10 UTC from IEEE Xplore. Restrictions apply.

Semantic Mapping: To set up the semantic mapping algorithm introduced in [13], we use a Detectron2 model [38], trained on the COCO dataset [39], to semantically segment RGB images. The segmented outputs are fused over time to construct a 3D semantic octomap with resolution $r_m = 1$ m. In our experiments, since we focus on a ground robot, we project this 3D map onto the X–Y plane, while considering height ranges relevant to the robot;

this gives rise to a 2D semantic class–based grid map. In doing so, we assume that no semantic objects are stacked vertically.

Exploitation-Exploration: The exploitation mode of Algorithm 2 is implemented using a standard A* search algorithm on the 2D grid map [1]. For exploration, we design an algorithm that randomly selects a next state $\mathbf{x}_{t+1}$ near a voxel with a non-singleton prediction set subject to (6), aiming to decrease its prediction set size. As discussed in Section III-B, however, any other exploration algorithm can be substituted.[3]

Conformalization of Maps: Next, we discuss our CP implementation described in Section III-A. Recall that the NCS in (4) requires evaluating the maximum over the set $\mathcal{P}$, which contains an infinite number of paths. Since this is impractical, following [24], we approximate the NCS using a finite subset $\hat{\mathcal{P}} \subset \mathcal{P}$ consisting of 10 paths between start and goal locations that are randomly selected in the calibration environment $\Omega_i \sim \mathcal{D}$. At test time, given $\Omega_{\text{test}} \sim \mathcal{D}$, we randomly select the start and goal positions for the task ψ. Also, to avoid generating a new calibration dataset for every Ω_{test}, we apply CP as described in Remark IV.4, with $\delta = 0.1$. Our calibration dataset consists of 50 environments.

Evaluation Metrics: We consider four metrics. (i) *Mapping success rate*, defined as the percentage of cases, where the following binary variable is 1: $S.R_{\text{map}} := \mathbf{1}(C_{0:H})$, where $C_{0:H} = \bigwedge_{\forall j \in \mathcal{J}_t, t \in [0:H]} (k_j^{\text{gt}} \in \mathcal{C}_j(\mathbf{m}_t))$ and k_j^{gt} is the ground truth label of cell j. In words, for a given test environment Ω_{test}, $S.R_{\text{map}} = 1$ if all constructed prediction sets $\mathcal{C}_j(\mathbf{m}_t)$ contain the true labels k_j^{gt} for all $j \in \mathcal{J}_t$ and $t \in [0:H]$. Due to Proposition IV.1, the average mapping success rate of Algorithm 2 across test environments drawn from $\mathcal{D}$ is expected to be at least $1-\alpha$ even in the presence of a sensor with limited FoV. (ii) *Mission success rate*, defined as the percentage of cases, where the binary variable $S.R_{\text{mission}} := \mathbf{1}(\mathbf{x}_{0:H} \models \psi)$ is 1. In words, given a test environment Ω_{test}, and task ψ_{test}, $S.R_{\text{mission}} = 1$ if the robot path $\mathbf{x}_{0:H}$ satisfies the safety constraints in (9) and $\mathbf{x}_H \in \mathcal{X}_g$. Note that if the sensor was omnidirectional, the mission success rate of Algorithm 2 should have been to be at least $1-\alpha$ due to Theorem IV.3; see our discussion in Section V-B. (iii) *Distance* traveled by the robot during deployment, which captures efficiency of the generated paths. (iv) *Proportion of the overall path used for exploration*, denoted as p_{expl}, quantifying how much of the trajectory was dedicated to reducing uncertainty in the environment. We report the average of these four metrics over 61 test environments. We also report the standard deviation of the binary variables $S.R_{\text{map}}$ and $S.R_{\text{mission}}$.

[3] A potential approach to handle cases in which the exploration algorithm fails to sufficiently reduce environmental uncertainty within a reasonable time is to find the largest $1-\alpha' < 1-\alpha$ for which a feasible path exists. This reduces the success rate to $1-\alpha'$ but allows the proposed method to generate a best-effort path.

TABLE I
COMPARATIVE EVALUATIONS

Planner	$1-\alpha$ (%)	$S.R_{\text{map}}$ (%)	$S.R_{\text{mission}}$ (%)	Path length (m)	p_{expl} (%)
Ours	95	93.36 ± 3.19	98.36 ± 1.63	74.7	19
	90	90.16 ± 3.81	96.72 ± 2.28	71.7	18.54
	85	86.89 ± 4.32	95.08 ± 2.77	69.72	11.2
UI	95	58.33	93.33	69.66	8.17
	90	41.67	76.67	66.14	4.65
	85	35	75	62.53	5.04
UA	NA	10.17	65.57	48.91	NA

TABLE II
EFFECT OF $|\hat{\mathcal{P}}|$

$1-\alpha$ (%)	n	$S.R_{\text{map}}$ (%)	$S.R_{\text{mission}}$ (%)	Path length (m)	p_{expl} (%)
90	1	90.33 ± 6.04	96.56 ± 2.34	63.727	5.59
	5	87.66 ± 4.12	89.22 ± 3.94	75.206	13.59
	10	90.16 ± 3.81	96.72 ± 2.28	71.7	18.54
	20	90 ± 3.79	100 ± 0	76.7338	18.81

B. Empirical Evaluation of the Proposed Planner

We evaluate our proposed method for three user-defined success rates: $1-\alpha \in \{95\%, 90\%, 85\%\}$ over 61 environments drawn from $\mathcal{D}$. The empirical results are reported in Table I. Observe that the empirical mapping success rates validate Prop. IV.1. It can be seen that for $1-\alpha = 95\%$, the empirical mapping success rate is $93.36\% < 95\%$. Such small deviations are due to the finiteness of the calibration dataset, the number of paths that are considered for calibration, and the number of validation environments; see [22] for more information on the influence of finiteness on the empirical success rates. We also observe that the empirical mission success rates validate Theorem IV.3. We note that we did not observe any mission failures due to limited FoV.[4] The path length increases with higher values of $1-\alpha$. This is expected, as higher $1-\alpha$ typically results in non-singleton prediction sets, which produce a more conservative assignment of labels to cells, as discussed in Section III-B. Consequently, our planner generates longer paths to account for the most conservative labels. Similarly, since higher $1-\alpha$ often leads to non-singleton prediction sets, the proportion of the overall path spent on exploration is also higher.

C. Effect of Number $|\hat{\mathcal{P}}|$ of Calibration Plans

As mentioned in Section V-A, we use a finite set $\hat{\mathcal{P}} \subset \mathcal{P}$ to estimate the NCS in (4). In what follows, we evaluate the performance of our algorithm for different sizes of $\hat{\mathcal{P}}$. Specifically, we use the same 50 calibration environments and the same 61 test environments described in Section V-A for $|\hat{\mathcal{P}}| = 1, 5, 10$, and 20. Notably, the empirical mapping success rates shown in Table II remain close to $1-\alpha$ for all considered $|\hat{\mathcal{P}}|$, while the

[4] A heuristic approach to prevent mission failures due to limited FoV is to have the robot spin at time t to collect measurements, or to restrict motions into unknown areas lying within its blind spots.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:01:10 UTC from IEEE Xplore. Restrictions apply.

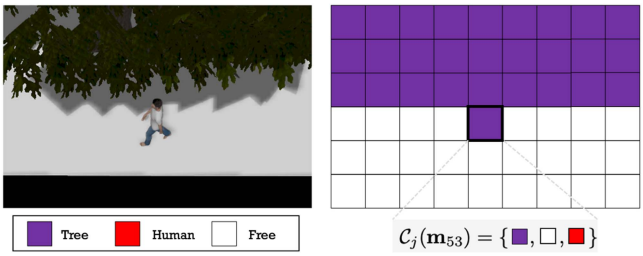


Fig. 4. Top view of the environment (left) and the corresponding map $\mathbf{m}_t$ (right). The most likely label in $\mathbf{m}_t$ for cell j, occupied by a human, is 'tree'. In contrast, the prediction set constructed by our method for that cell mitigates this mapping error by including 'tree', 'human', and 'free-space'.

standard deviation of $S.R_{\text{map}}$ tends to decrease, as $|\hat{\mathcal{P}}|$ increases. We emphasize that this observation may be case-specific; in theory, the mapping success rate is guaranteed to be at least $1-\alpha$ only when $\hat{\mathcal{P}} = \mathcal{P}$. As $|\hat{\mathcal{P}}|$ increases, the NCS becomes larger, resulting in larger, non-singleton prediction sets. This increased conservativeness is reflected in higher (though not necessarily monotonic) mission completion rates, longer paths, and larger exploration proportion.

D. Comparative Experiments

Baselines: We compare our planner against the following baselines. (i) *Uncertainty-agnostic (UA) Planner:* This planner assigns to each cell j the most-likely label as per the PMFs $p_t(m_j)$, i.e., the label assigned to cell j at time t is $k_j = \arg\max_{k\in\mathcal{K}} p_t(m_j)$. Using this label assignment, the planner employs A$*$ to compute paths that complete ψ. We select this baseline to illustrate the importance of designing planners that are aware of mapping uncertainty. (ii) *Uncertainty-informed (UI) Planner:* The second baseline differs from our planner only in the construction of prediction sets. Specifically, the predictions sets are built heuristically by treating the PMFs $p_t(m_j)$ as true probabilities. Thus, the prediction sets are constructed as $\mathcal{C}_j(\mathbf{m}_t) = \{\pi_1, \ldots, \pi_k\}$, where $k = \sup\{k' : \sum_{i=1}^{k'} p_t(m_j = \pi_i) < 1-\alpha\} + 1$, and π is the permutation of $\mathcal{K}$ that sorts the classes from most likely to least likely, i.e. $p_t(m_j = \pi_{n-1}) > p_t(m_j = \pi_n)$. We select this baseline to illustrate the importance of calibrating uncertainty metrics provided by mapping algorithms.

Comparative Results: Table I shows the performance of both baselines using the evaluation metrics discussed in Section V-A. To ensure a fair comparison, both baselines are evaluated on the same 61 test environments, and their corresponding reach-avoid tasks used for our proposed method. Notice that the UI planner outperforms the UA planner in terms of mission and mapping success rates, as it accounts for mapping uncertainty. However, our method outperforms the UI baseline because the UI prediction sets—unlike ours—are constructed heuristically.

As seen in Table I, our approach leads to a significant improvement in mapping and mission success rates. Our empirical evaluation shows that most baseline failures occur due to partial occlusions in the environment. An example is illustrated in Fig. 4, where a human is partially occluded by trees. The cell associated with the human is often labeled as tree' by the UA planner, leading to safety violations because the human' label has stricter safety constraints than the tree' label. In this case, by the time the UA planner detects the human, a safety constraint has already been violated. Similarly, the UI planner builds a singleton prediction set containing only the 'tree' label. In contrast, our approach constructs a prediction set that includes 'tree', 'human', and 'free-space', enabling the planner to follow a conservative path and maintain safety. This results in higher mission and mapping success rates, though at the cost of longer paths, as shown in Table I.

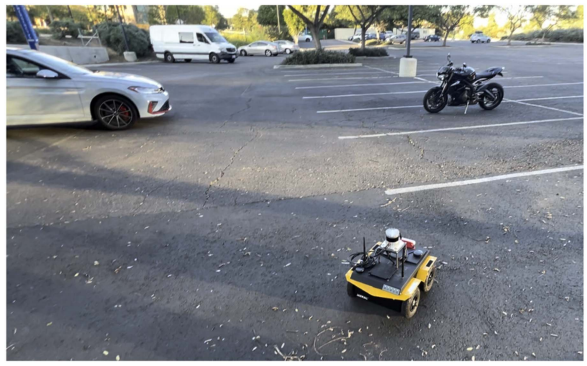


Fig. 5. Hardware demonstration of the proposed method.

E. Hardware Experiments

In this section, we demonstrate robot behaviors with respect to various values for $1-\alpha$ in real-world settings. Specifically, we consider a Clearpath Jackal equipped with an Intel RealSense D455 camera and a Detectron2 segmentation model [38]. The robot operates in environments with humans, motorcycles, cars, and tree (see Fig. 5). The class-based safety constraints are: $d_{\text{person}} = 1.5$ m, $d_{\text{car}} = 1.25$ m, $d_{\text{motorcycle}} = 1$ m, and $d_{\text{tree}} = 1$ m. Similar to the simulation results, we observed an increasing trend in path length as $1-\alpha$ increases. Specifically, the total traveled distance for $1-\alpha = 90\%, 95\%$, and 99% was 24.38 m, 24.92 m, and 25.78 m, respectively. We note that our method was applied using calibration data described in Section V-A, which induces a distribution shift. Evaluation of our approach in OOD settings in terms of mapping and mission success rates is provided in Section V-F.

F. Evaluation in Out-of-Distribution (OOD) Scenarios

Finally, we evaluated our method in OOD environments. For calibration, we use the same 50 calibration environments sampled from $\mathcal{D}$ mentioned in Section V-A, with $|\hat{\mathcal{P}}| = 10$. In what follows, we set $1-\alpha = 90\%$. We consider the case where the test environments are sampled from a distribution $\mathcal{D}'$ where, unlike $\mathcal{D}$, the trees vary in number from 1 to 10, and are placed randomly. In addition, the environments also simulated the lighting conditions of dusk and dawn instead of midday used in $\mathcal{D}$. We observed a drop in mapping success rates for $1-\alpha = 90\%$ from $90.16\% \pm 3.81\%$ to $80.33\% \pm 5.09\%$. The mission success rates also dropped from $96.72\% \pm 2.28\%$ to $93.44\% \pm 3.17\%$, albeit still being greater than $1-\alpha$. We empirically observed that changes in the number and spatial distribution of the trees have a more significant impact than the lightning conditions on the performance of the algorithm.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:01:10 UTC from IEEE Xplore. Restrictions apply.

VI. Conclusions and Future Work

In this paper, we presented a novel semantic planner for unknown environments. Unlike related works, our method designs probabilistically correct paths in the presence of environmental uncertainty, stemming from imperfect perception, without any assumptions on sensor models or noise characteristics. Future work will focus on: (i) designing active exploration algorithms that directly target reducing the size of prediction sets; (ii) accounting for OOD scenarios by leveraging robust CP methods; and (iii) directly quantifying sensor (rather than mapping) uncertainty.

References

[1] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, Jul. 1968.

[2] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.

[3] L. Gionfrida, C. Wang, L. Gan, M. Chli, and L. Carlone, "Computer and robot vision: Past, present, and future [TC spotlight]," *IEEE Robot. Automat. Mag.*, vol. 31, no. 3, pp. 211–215, Sep. 2024.

[4] J. A. Placed et al., "A survey on active simultaneous localization and mapping: State of the art and new frontiers," *IEEE Trans. Robot.*, vol. 39, no. 3, pp. 1686–1705, Jun. 2023.

[5] Y. Kantaros, S. Kalluraya, Q. Jin, and G. J. Pappas, "Perception-based temporal logic planning in uncertain semantic maps," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2536–2556, Aug. 2022.

[6] J. Fu, N. Atanasov, U. Topcu, and G. J. Pappas, "Optimal temporal logic planning in probabilistic semantic maps," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 3690–3697.

[7] V. Vasilopoulos et al., "Reactive semantic planning in unexplored semantic environments using deep perceptual feedback," *IEEE Robot. Automat. Lett.*, vol. 5, no. 3, pp. 4455–4462, Jul. 2020.

[8] G. Malczyk, M. Kulkarni, and K. Alexis, "Semantically-driven deep reinforcement learning for inspection path planning," *IEEE Robot. Automat. Lett.*, vol. 10, no. 7, pp. 7206–7213, Jul. 2025.

[9] A. Pal, Y. Qiu, and H. Christensen, "Learning hierarchical relationships for object-goal navigation," in *Proc. Conf. Robot Learn.*, 2021, pp. 517–528.

[10] Z. Ravichandran, L. Peng, N. Hughes, J. D. Griffith, and L. Carlone, "Hierarchical representations and explicit memory: Learning effective navigation policies on 3D scene graphs using graph neural networks," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 9272–9279.

[11] G. Georgakis, B. Bucher, A. Arapin, K. Schmeckpeper, N. Matni, and K. Daniilidis, "Uncertainty-driven planner for exploration and navigation," in *Proc. Int. Conf. Robot. Automat.*, 2022, pp. 11295–11302.

[12] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic slam," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 1722–1729.

[13] A. Asgharivaskasi and N. Atanasov, "Semantic OcTree mapping and Shannon mutual information computation for robot exploration," *IEEE Trans. Robot.*, vol. 39, no. 3, pp. 1910–1928, Jun. 2023.

[14] Y. Tian, Y. Chang, F. H. Arias, C. Nieto-Granda, J. P. How, and L. Carlone, "Kimera-multi: Robust, distributed, dense metric-semantic SLAM for multi-robot systems," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2022–2038, Aug. 2022.

[15] E. Psomiadis, and P. Tsiotras, "HCOA∗: Hierarchical Class-Ordered A∗ for Navigation in Semantic Environments," IEEE Robotics and Automation Letters, vol. 10, no. 10, p.

[16] J. Wilson et al., "LatentBKI: Open-dictionary continuous mapping in visual-language latent spaces with quantifiable uncertainty," *IEEE Robot. Automat. Lett.*, vol. 10, no. 4, pp. 3102–3109, Apr. 2025.

[17] V. Renganathan, S. Safaoui, A. Kothari, B. Gravell, I. Shames, and T. Summers, "Risk bounded nonlinear robot motion planning with integrated perception & control," *Artif. Intell.*, vol. 314, 2023, Art. no. 103812.

[18] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 723–730.

[19] L. Palmieri, T. P. Kucner, M. Magnusson, A. J. Lilienthal, and K. O. Arras, "Kinodynamic motion planning on Gaussian mixture fields," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 6176–6181.

[20] Q. H. Ho, Z. N. Sunberg, and M. Lahijanian, "Gaussian belief trees for chance constrained asymptotically optimal motion planning," in *Proc. Int. Conf. Robot. Automat.*, 2022, pp. 11029–11035.

[21] M. Minderer et al., "Revisiting the calibration of modern neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 15682–15694.

[22] A. N. Angelopoulos and S. Bates, "A gentle introduction to conformal prediction and distribution-free uncertainty quantification," 2022, *arXiv: 2107.07511*.

[23] A. Ray, C. Bradley, L. Carlone, and N. Roy, "Task and motion planning in hierarchical 3D scene graphs," 2024, *arXiv:2403.08094*.

[24] A. Dixit, Z. Mei, M. Booker, M. Storey-Matsutani, A. Z. Ren, and A. Majumdar, "Perceive with confidence: Statistical safety assurances for navigation with learning-based perception," in *Proc. 8th Conf. Robot Learn.*, vol. 270, 2025, pp. 2517–2541.

[25] L. Lindemann, M. Cleaveland, G. Shim, and G. J. Pappas, "Safe planning in dynamic environments using conformal prediction," *IEEE Robot. Automat. Lett.*, vol. 8, no. 8, pp. 5116–5123, Aug. 2023.

[26] S. Sheng, P. Yu, D. Parker, M. Kwiatkowska, and L. Feng, "Safe POMDP online planning among dynamic agents via adaptive conformal prediction," *IEEE Robot. Automat. Lett.*, vol. 9, no. 11, pp. 9946–9953, Nov. 2024.

[27] A. Doula, T. Güdelhöfer, M. Mühlhäuser, and A. S. Guinea, "Conformal prediction for semantically-aware autonomous perception in urban environments," in *Proc. 8th Conf. Robot Learn.*, in Proceedings of Machine Learning Research Series, P. Agrawal, O. Kroemer, and W. Burgard, Eds., Nov. 2025, vol. 270, pp. 2629–2654. [Online]. Available: https://proceedings.mlr.press/v270/doula25a.html

[28] A. Z. Ren, et al., "Robots that ask for help: Uncertainty alignment for large language model planners," in *Proc. 7th Annu. Conf. Robot Learn.*, 2023.

[29] J. Wang, G. He, and Y. Kantaros, "Probabilistically correct language-based multi-robot planning using conformal prediction," *IEEE Robot. Automat. Lett.*, vol. 10, no. 1, pp. 160–167, Jan. 2025.

[30] J. Lekeufack, A. N. Angelopoulos, A. Bajcsy, M. I. Jordan, and J. Malik, "Conformal decision theory: Safe autonomous decisions from imperfect predictions," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2024, pp. 11668–11675.

[31] J. Sun, Y. Jiang, J. Qiu, P. Nobel, M. J. Kochenderfer, and M. Schwager, "Conformal prediction for uncertainty-aware planning with diffusion dynamics model," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, vol. 36, pp. 80324–80337 .

[32] L. Janson, E. Schmerling, A. Clark, and M. Pavone, "Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions," *Int. J. Robot. Res.*, vol. 34, no. 7, pp. 883–921, 2015.

[33] R. P. de Figueiredo, J. le Fevre Sejersen, J. G. Hansen, M. Brandão, and E. Kayacan, "Real-time volumetric-semantic exploration and mapping: An uncertainty-aware approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 9064–9070.

[34] A. N. Angelopoulos and S. Bates, "Conformal prediction: A gentle introduction," *Foundations Trends Mach. Learn.*, vol. 16, no. 4, pp. 494–591, 2023.

[35] V. Vovk, "Conditional validity of inductive conformal predictors," in *Proc. Asian Conf. Mach. Learn.*, 2012, pp. 475–490.

[36] "Experiment_Video," 2026. [Online]. Available: https://vimeo.com/1161521936?fl=ip&fe=ec

[37] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Sci. Robot.*, vol. 7, no. 66, 2022, Art. no. eabm6074.

[38] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," 2019. [Online]. Available: https://github.com/facebookresearch/detectron2

[39] T. Lin et al., "Microsoft COCO: Common objects in context," *Fleet, Comput. Vis.*, vol. 8693, 2014.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:01:10 UTC from IEEE Xplore. Restrictions apply.