

Object- and Task-Aware Active Perception for Safe Navigation of Mobile Manipulators in Semi-Static Environments

Jingxing Qian^{*,1}, Xintong Du^{*,1}, Jun Yang², Siqi Zhou^{3,4}, Steven L. Waslander⁵, and Angela P. Schoellig³

Abstract—Mobile manipulators must navigate safely in cluttered, partially unknown environments across logistics, manufacturing, and service applications. Semi-static changes between robot visits can lead to outdated scene knowledge and increased collision risk. We propose an active perception framework that enables a high degree-of-freedom (DoF) mobile manipulator to inspect uncertain or safety-critical regions with its hand camera during navigation. A probabilistic, object- and task-aware next-best-view (NBV) formulation prioritizes observations near the desired path and around uncertain or potentially changing objects. The perception objective is integrated into hierarchical-task model predictive control (HTMPC), exploiting kinematic redundancy for viewpoint adjustment while preserving the higher priority of base-trajectory tracking. The system unifies navigation, map maintenance, and active perception in a closed loop. Real-world tests show improved map consistency, reduced task-dependent map entropy, and lower navigation risk than passive or non-task-aware baselines. Additional simulation and real-world sequences demonstrate responses to controlled semi-static scene changes.

I. INTRODUCTION

Mobile manipulators have gained increasing interest in a wide range of applications, including household assistance, logistics, and industrial automation. In these settings, robots are often required to execute navigation tasks across large workspaces over extended periods of time [1]. However, many deployment sites are cluttered, partially unknown, and semi-static, where objects may appear, disappear, or shift between robot visits. Clutter introduces frequent occlusions, blind corners, and narrow passages, while semi-static scene changes cause previously observed regions to become outdated. These challenges are particularly pronounced when the robot must perceive the environment through a limited field of view (FoV). In such cases, occlusions and blind spots can prevent changes from being detected until the robot is in close proximity, leaving new obstacles in previously traversable areas undetected. This can lead to delayed reactions, increased collision risk, and reduced navigation efficiency.

To operate safely and robustly in such cluttered, semi-static environments, a mobile manipulator must not only

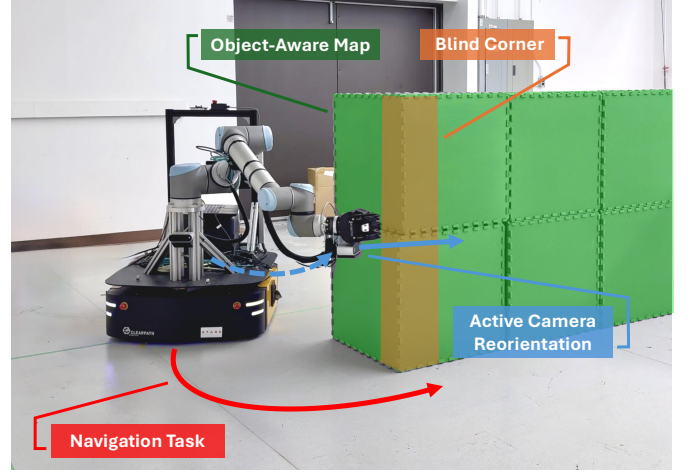


Fig. 1: A snapshot of the proposed active perception and navigation framework for high-degree-of-freedom (DoF) mobile manipulators in action. Equipped with a hand camera, the robot operates in cluttered, semi-static environments. It uses its kinematic redundancy to reposition the camera around obstacles and observe uncertain or occluded regions near its navigation path. See the multimedia attachment or online videos⁶ for demonstrations.

navigate toward its desired goal but also maintain an up-to-date, consistent understanding of the scene over time. However, this requirement is not well supported by most existing navigation pipelines, which typically assume a static world with an offline-constructed map for downstream planning and control [2], or only track short-term, highly dynamic objects between consecutive frames. Recent research on long-term and semi-static mapping has made important progress toward maintaining consistent environment representations across repeated traversals [3], [4]. By incorporating semantic and object-level reasoning, these approaches can detect scene inconsistencies and accommodate long-term changes, thereby improving map quality and robustness over time. Nevertheless, such methods remain fundamentally passive: they depend on observations collected along the robot's nominal trajectory, and thus can only update the regions that the robot happens to observe. In cluttered environments, this is often insufficient, especially in occluded regions where up-to-date scene knowledge is most needed.

This limitation is especially critical because not all map regions affect navigation equally. Outdated knowledge in task-relevant and safety-critical regions, such as areas along the planned path, narrow passages, and blind corners, has a disproportionate impact on the robot's ability to move safely and efficiently. As seen in Figure 1, when the robot makes a sharp turn around the boxes, it is most important for the robot to observe the region around its navigation path behind

^{*}Jingxing Qian and Xintong Du contributed equally.

¹University of Toronto Institute for Aerospace Studies (UTIAS) and the Vector Institute for Artificial Intelligence, Canada. Emails: jingxing.qian@robotics.utias.utoronto.ca, xintong.du@robotics.utias.utoronto.ca.

²Epson Canada Ltd., Toronto, ON L3R 6G3, Canada. Email: Jun.Yang@ea.epson.com.

³Learning Systems and Robotics Lab at the Technical University of Munich and the Munich Institute for Robotics and Machine Intelligence (MIRMI), Germany. Email: angela.schoellig@tum.de.

⁴School of Computing Science, Faculty of Applied Sciences, Simon Fraser University, Burnaby, BC, Canada. Email: siqi@sfu.ca.

⁵Institute for Aerospace Studies (UTIAS) and Robotics Institute, University of Toronto, Canada. Email: steven.waslander@robotics.utias.utoronto.ca.

⁶Video demonstrations: <https://tinyurl.com/nbv-for-mm-nav>.

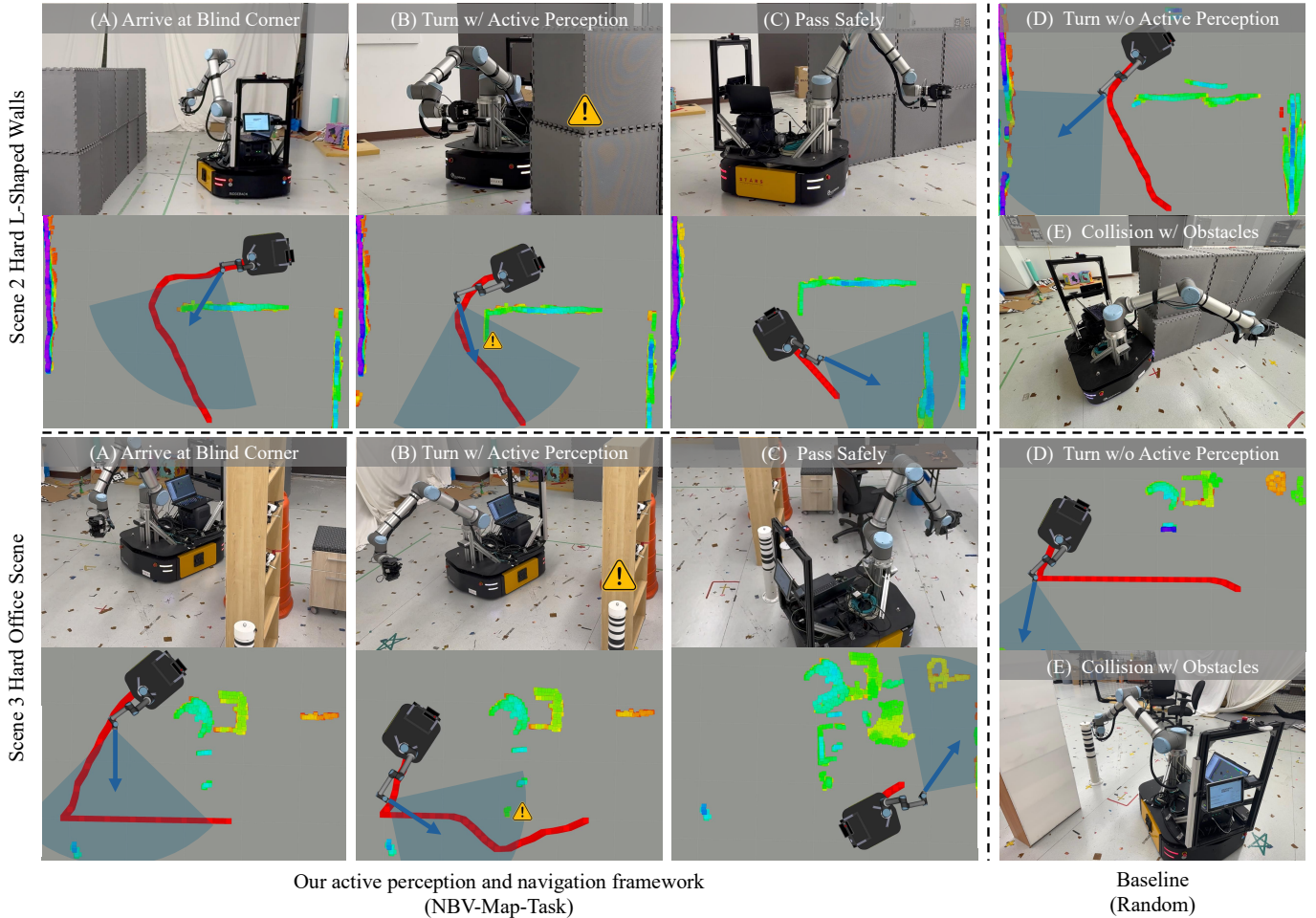


Fig. 2: Qualitative demonstration of the proposed active perception and navigation framework for mobile manipulation in action (left) compared with a baseline method that randomly points the hand camera (right). Red curves represent the planned base paths. Shaded blue regions represent the hand camera’s FoV. Yellow exclamation marks highlight hidden hazards such as a blind corner (top) and a hidden pole (bottom).

the orange-shaded blind corner, which is not immediately observable by the fixed camera on the robot base. An outdated or incomplete map in these regions can lead to overly conservative behavior, unnecessary replanning, getting stuck, or collisions. Therefore, map maintenance is particularly valuable in regions where it improves the robot’s ability to make safe task decisions. In this work, we refer to this principle as *task-awareness*: observations should be prioritized according to their relevance to the robot’s current navigation task, especially when they reduce uncertainty in map regions that affect safe progress toward the goal.

Active perception provides a means to acquire these observations during task execution. Related approaches couple perception and motion planning for control and collision avoidance [5], [6]. Next-best-view (NBV) planning has also been studied for exploration [7], object seeking [8], inspection [9], pose estimation [10], and grasping [11]. A camera mounted on the manipulator hand provides a natural mechanism for viewpoint control. However, many NBV methods target short-horizon operation in static scenes [12], [13] or geometric completeness [14], [15]. They generally do not jointly consider long-term scene changes, object-level map consistency, and relevance to the current navigation task.

For semi-static environments, this omission is particularly important because scene changes often occur at the level of objects: furniture may be moved, shelves may be added, or obstacles may partially block previously traversable space. A partial mismatch at the visible edge of a cabinet can indicate that the cabinet has moved, making its previously mapped region worth checking again. As a result, identifying which objects or occupied structures are likely to have changed and which of them matter more for the robot’s navigation task is critical for effective map maintenance and safe task execution. This motivates an object-aware active perception strategy in which viewpoint selection is informed by uncertainty in the maintained semi-static map, especially when such uncertainty may affect path clearance. Task awareness then determines whether this potential change matters to the next maneuver.

Moreover, because the hand camera is mounted on a manipulator, viewpoint selection cannot be treated independently from navigation. The robot must realize informative camera motions without compromising progress on its primary navigation task. We therefore use hierarchical-task model predictive control (HTMPC) as the mechanism for enforcing this priority structure and realizing the selected viewpoints through the robot’s kinematic redundancy [16], [17].

In this work, we aim to enable safe mobile manipulation in cluttered, semi-static environments over extended periods of time through integrated active perception and whole-body control. Specifically, we develop an object- and task-aware active perception framework that combines the planned navigation path with uncertainty in the maintained semi-static map. Our approach enables a mobile manipulator to exploit its kinematic redundancy to direct a hand camera toward uncertain, occluded, and safety-critical regions while continuing to execute its primary navigation task. By actively updating the map where it matters most, the robot can better detect and adapt to incremental scene changes and improve map consistency in task-relevant areas. We deploy the system on a real mobile manipulator and evaluate reconstruction accuracy, task-dependent map entropy, and navigation safety against passive and task- or object-unaware baselines. Figure 2 illustrates the resulting behavior in cluttered scenes. Our main contributions are as follows:

- We formulate a probabilistic next-best-view (NBV) objective that uses task-dependent map entropy and path-conditioned grid weights to prioritize uncertain or potentially changing regions along the planned path.
- We introduce a hierarchical-task control framework that assigns higher priority to navigation than to active perception within a two-stage optimization problem. This allows the robot to exploit its kinematic redundancy to gather informative observations while bounding the base-tracking cost during camera-task optimization.
- We demonstrate the system on a mobile manipulator with a hand camera in cluttered indoor environments. In the tested static scenes, the proposed system reduces task-dependent map entropy, improves map consistency, and lowers navigation risk compared with passive or non-task-aware baselines. Additional scene-change sequences demonstrate the closed-loop map-maintenance behavior.

II. RELATED WORK

A. Semantic and Object-Aware Mapping

Early mapping systems rely on sparse feature points [18] or dense volumetric grids such as occupancy grids [19] and signed or Euclidean distance fields (SDFs/EDFs) [20] to construct an accurate geometric map. However, purely geometric maps lack semantic context, limiting their expressiveness in complex environments. More recent work has incorporated semantic and object-level knowledge into geometric maps. Early approaches introduced objects into mapping by using predefined CAD models [21]. Subsequent object-aware methods further decompose the scene into semantic or object submaps on-the-fly [22]–[24], allowing higher-level reasoning about object relationships and scene structure for downstream planning tasks [25], [26].

Despite these advances, most semantic and object-aware mapping systems continue to prioritize geometric accuracy under a static-world assumption. When deployed over long time horizons in real-world semi-static environments, these methods can accumulate inconsistencies, leading to map corruption, degraded localization, and increased safety risks.

B. Change Detection in Semi-Static Environments

To relax the static-world assumption, several strategies have been proposed to handle scene dynamics. A common approach identifies potentially dynamic objects based on prior semantic knowledge and treats them as outliers [27], [28]. While efficient, this strategy can lead to information loss and instability when large parts of the scene have changed [29]. Another strategy jointly tracks the robot and objects [22], [30], [31], but typically relies on motion consistency across consecutive frames, making them less effective for semi-static environments, where changes occur between robot visits.

Detecting semi-static changes is critical for long-term autonomy, but has received less attention. Some recent approaches estimate object-level consistency based on overlap between observations and mapped objects [3], [32]. Other approaches leverage learned representations or scene graphs to predict object changes [33], [34], use Bayesian formulations to propagate object consistency [4], [35], and jointly optimize robot state and object consistency [36].

While these methods improve long-term map consistency, they are typically evaluated in open-loop mapping scenarios and rely on passive observations. As a result, changes occurring in occluded or task-critical regions may remain undetected until they lead to failure. An open question is how to couple probabilistic object-level change estimates with online viewpoint selection and control for safe navigation.

C. Active Perception in Robotics

Active perception deliberately selects sensor viewpoints to acquire observations that improve a robotic task [37]. For exploration and reconstruction, NBV planners commonly seek views that reveal unmapped space or reduce map uncertainty [38], [39]. Naazare et al. [9] extend this principle to exploration and inspection with an arm-mounted RGB-D camera, combining information gain with coverage of user-specified regions of interest. These approaches establish the utility of movable sensors, while the choice of which regions to inspect depends on the task being performed.

For navigation, Fisher-information methods select observations that improve localization [14], [40]. The Fisher Information Field [41], [42] represents the information available for visual localization and supports evaluation of candidate poses during motion planning. Learning-based methods instead predict informative viewpoints from data for localization [12], [13] or exploration [15]. Our objective uses map uncertainty and path relevance to choose observations of obstacles and free space that affect navigation.

Visibility-aware and risk-averse planners also account for sensing limitations and path-relevant uncertainty when planning motion at the cost of navigation efficiency [43], [44]. Our focus is maintaining object-level consistency across repeated visits and realizing informative hand-camera views under a navigation-first task hierarchy.

D. Perception-based Mobile Manipulation

Perception-based control frameworks, in which onboard sensor data continuously inform motion planning and control,

have enabled mobile manipulators to operate in changing environments for tasks such as reactive grasping and obstacle avoidance [45]–[47]. In all these works, however, the coupling is one-directional: the robot perceives to inform control, but sensor viewpoints are determined passively by the task-driven motion rather than actively optimized. Active perception has been explored in mobile manipulation for specific subtasks such as object recognition [48] and grasp pose detection [49]. SPIN [50] further learns coordinated perception, interaction, and navigation from onboard vision without constructing an environment map. Our setting emphasizes persistent object-map maintenance: observations acquired during navigation must reveal changes to a scene mapped on earlier visits, and the resulting map must inform subsequent motion.

Hierarchical task control offers a principled way to integrate active perception into whole-body motion planning for kinematically redundant robots. By establishing a strict order among tasks, the hierarchical task control framework leverages the robot’s redundancy such that the lower-priority tasks are solved only in the null space of higher-priority ones [51], [52], avoiding conflicting motions inherent in the tasks. HTMPC instantiates this idea in a model predictive control setting through nonlinear lexicographic optimization, which enables strictly ordered sequential mobile manipulation tasks [16], [17]. In this work, we employ HTMPC with active perception as an explicit task in the hierarchy, secondary to the navigation task, resolving the inherent perception-control objective conflicts and creating a truly bidirectional perception-control loop: the robot plans its action based on the latest perceived map, and simultaneously optimizes its sensor placement to actively perceive the surrounding environment. This enables safe whole-body navigation in unstructured, semi-static environments.

III. PRELIMINARIES

A. Object-Aware Semi-Static Mapping

We consider semi-static environments in which most objects remain stationary over short time horizons, but may appear, disappear, or change location between repeated robot visits. In such settings, maintaining a globally consistent map requires not only estimating scene geometry from observations, but also reasoning about whether previously mapped objects are still valid. To achieve this, we build on the probabilistic object-level change detection framework in POCD [4], which represents the scene as an object library $\mathcal{O} = \{\mathbf{O}_i\}_{i=1\dots I}$ and explicitly models the temporal and geometric consistency of each object. Each mapped object $\mathbf{O}_i$ forms an object-level submap containing:

- a 3D position and a planar yaw angle in the global frame,
- an EDF voxel-grid submap $\mathcal{M}_i$ in the global frame,
- a state distribution $p(l_i, v_i)$, where $l_i \in \mathbb{R}$ models the object’s geometric change and $v_i \in [0, 1]$ models the object’s consistency, with larger values indicating a higher probability that the mapped object remains valid.

This object-centric representation allows the system to reason about scene changes at the object level, which is more robust than treating map cells independently under partial observations and repeated traversals.

Following [4], the state distribution of object $\mathbf{O}_i$ at time step k is parameterized as the product of a Gaussian distribution for the continuous geometric-change variable l_i and a Beta distribution for the bounded consistency variable v_i :

$$\begin{aligned} p(l_i, v_i) &:= q(l_i, v_i \mid \mu_{i,k}, \sigma_{i,k}, \alpha_{i,k}, \beta_{i,k}) \\ &:= \mathcal{N}(l_i \mid \mu_{i,k}, \sigma_{i,k}^2) \text{Beta}(v_i \mid \alpha_{i,k}, \beta_{i,k}). \end{aligned} \quad (1)$$

Here, $\mu_{i,k}$ and $\sigma_{i,k}^2$ represent the current estimate and variance of the geometric change, while $\alpha_{i,k}$ and $\beta_{i,k}$ parameterize the consistency belief. The expectation

$$\mathbb{E}[v_i] = \frac{\alpha_{i,k}}{\alpha_{i,k} + \beta_{i,k}} \quad (2)$$

serves as the *object consistency estimate*. It has been used in prior works for map maintenance and for downstream planning and control to indicate potentially changed objects around which the robot should behave cautiously [8], [53]. In this work, we additionally use the variance

$$\text{Var}[v_i] = \frac{\alpha_{i,k}\beta_{i,k}}{(\alpha_{i,k} + \beta_{i,k})^2(\alpha_{i,k} + \beta_{i,k} + 1)} \quad (3)$$

as an uncertainty measure for the object’s consistency. Thus, $\mathbb{E}[v_i]$ captures the expected validity of the mapped object, while $\text{Var}[v_i]$ captures uncertainty in that validity estimate.

For each matched object-observation pair, the measurement consists of two parts. First, the geometric consistency $\Delta_{i,k}$ is computed as the difference between the object and observation submaps. This term is modeled as an inlier when the object remains consistent with its current estimate, and as an outlier when the object has changed. The corresponding likelihood is written as the Gaussian-Uniform mixture [4], [54]

$$\begin{aligned} p(\Delta_{i,k} \mid l_i, v_i) &:= v_i \mathcal{N}(\Delta_{i,k} \mid l_i, \tau^2) \\ &\quad + (1 - v_i) \mathcal{U}(\Delta_{i,k} \mid -\Delta_{\max}, \Delta_{\max}), \end{aligned} \quad (4)$$

where τ^2 is the inlier measurement variance and $\Delta_{\max}$ bounds the outlier range. Second, a binary semantic label $s_{i,k} \in \{0, 1\}$ provides a prior on whether the observed object is likely static or likely changing. This term enters the update as a weighted Bernoulli likelihood factor:

$$p(s_{i,k} \mid v_i) := \text{Bernoulli}(s_{i,k} \mid v_i)^{\kappa_{i,k}}. \quad (5)$$

Here, $\kappa_{i,k} \geq 0$ controls the relative weight of semantic evidence. The Bayesian update combines the new geometric and semantic evidence with the previous state estimate. Writing the prior at time step k as $q(l_i, v_i \mid \mu_{i,k}, \sigma_{i,k}, \alpha_{i,k}, \beta_{i,k})$, the posterior at time step $k + 1$ satisfies

$$\begin{aligned} &q(l_i, v_i \mid \mu_{i,k+1}, \sigma_{i,k+1}, \alpha_{i,k+1}, \beta_{i,k+1}) \\ &\propto p(\Delta_{i,k} \mid l_i, v_i) p(s_{i,k} \mid v_i) \\ &\quad \times q(l_i, v_i \mid \mu_{i,k}, \sigma_{i,k}, \alpha_{i,k}, \beta_{i,k}). \end{aligned} \quad (6)$$

This posterior is obtained by multiplying the prior with the geometric-consistency and semantic-consistency likelihoods. It can then be approximated in closed form by matching moments with a single Gaussian-Beta distribution of the form in (1) [4]. Operationally, matched observations that agree with the current object model typically increase the consistency estimate and reduce its uncertainty, while inconsistent or missing

observations penalize consistency and increase uncertainty. If $\mathbb{E}[v_i]$ falls below a threshold, the object is treated as changed and reconstructed in the maintained map.

B. Fisher Information and Differential Entropy for NBV

Next-best-view (NBV) planning selects sensing actions that are expected to reduce uncertainty relevant to a downstream task. A common way to formalize this idea is through information-theoretic metrics that quantify how informative a candidate observation is with respect to an unknown quantity of interest. Among these metrics, Fisher information provides a local measure of observation sensitivity [55].

Consider a generic observation model

$$Y \mid x \sim p(\cdot \mid x), \quad (7)$$

where $Y \in \mathbb{R}$ is an observable random variable and $x \in \mathbb{R}^n$ is the latent quantity of interest. This generic notation can represent different active-perception quantities, such as a robot pose, map state, or scene parameter. In Section V-C, we instantiate this template with the local robot base position used in the EDF likelihood. The Fisher score is defined as

$$V = \nabla_x \log p(Y \mid x), \quad (8)$$

and the Fisher Information Matrix (FIM) is defined as

$$\begin{aligned} \mathcal{F}(x) &= \mathbb{E}[VV^T] \\ &= \mathbb{E}[(\nabla_x \log p(Y \mid x))(\nabla_x \log p(Y \mid x))^T]. \end{aligned} \quad (9)$$

Under standard regularity conditions, the score has zero mean, so this expression coincides with the covariance of V . The FIM quantifies how sensitive the observation model is to changes in the latent quantity of interest, and is therefore a measure of local informativeness. However, NBV also requires comparing how much uncertainty remains under the current belief. For this reason, differential entropy is commonly used as a scalar measure of estimation uncertainty [56]. Under a local Gaussian approximation, the covariance of the estimate of x can be approximated by $\mathcal{F}(x)^{-1}$. The corresponding differential entropy is

$$h_e(\mathcal{F}(x)) = \frac{1}{2} \log \left[(2\pi e)^n \det(\mathcal{F}(x)^{-1}) \right]. \quad (10)$$

Thus, under the local Gaussian approximation, uncertainty reduction can be measured by the decrease in the entropy induced by $\mathcal{F}(x)^{-1}$.

In active perception, information gain measures the reduction from prior to posterior uncertainty, while conventional expected information gain averages this reduction over future observations [39], [42]. Our approximate EIG measures the expected reduction in task-dependent map entropy over normalized grid-cell and rollout-time weights, using two representative observation outcomes per candidate view. The full construction is given in Section V-A–Section V-F. This objective ranks viewpoints by their predicted information gain. Realizing the selected view also requires coordinating camera and base motion, which motivates the task hierarchy introduced next.

C. Hierarchical-Task Model Predictive Control

The HTMPC framework provides a principled way to coordinate multiple control objectives with strict task priorities over a finite prediction horizon. This is particularly useful for high-DoF mobile manipulators, where secondary objectives such as camera reorientation for active perception should exploit the robot's kinematic redundancy without degrading the primary navigation objective. For example, inspecting a blind corner may require the hand camera to move around an obstacle while the base follows its path. The controller must coordinate these motions according to their task priorities.

Given an ordered set of tasks $[\tau_1, \tau_2, \dots, \tau_L]$ in decreasing order of priority, HTMPC is formulated at each control time step t_k as a lexicographic optimal control problem over a finite prediction time interval $\mathbb{T}_k := [t_k, t_k + T_{\text{mpc}}]$,

$$\text{lex min}_{\mathbf{x}(\cdot), \mathbf{u}(\cdot)} [\mathcal{J}_1, \mathcal{J}_2, \dots, \mathcal{J}_L] \quad (11a)$$

$$\text{s.t. } \dot{\mathbf{x}}(t) = \mathbf{f}_{mdl}(\mathbf{x}(t), \mathbf{u}(t)), \quad \forall t \in \mathbb{T}_k, \quad (11b)$$

$$\mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) \leq \mathbf{0}, \quad \forall t \in \mathbb{T}_k, \quad (11c)$$

$$\mathbf{x}(t) \in \mathcal{X}, \mathbf{u}(t) \in \mathcal{U}, \quad \forall t \in \mathbb{T}_k, \quad (11d)$$

$$\mathbf{x}(t_k) = \mathbf{x}_k, \quad (11e)$$

where t is the continuous time variable, $\mathbf{x}(\cdot) : \mathbb{R} \rightarrow \mathbb{R}^{n_x}$ and $\mathbf{u}(\cdot) : \mathbb{R} \rightarrow \mathbb{R}^{n_u}$ are continuous state and input trajectories, and $\mathbf{x}_k = \mathbf{x}(t_k)$ is the current state. Each $\mathcal{J}_l$ is the cost associated with task τ_l , and the list of cost functions is minimized in lexicographic order through lex min , subject to robot dynamics $\mathbf{f}_{mdl}$ and state- and input-dependent constraints $\mathbf{g}$, such as safety constraints. $\mathcal{X}$ and $\mathcal{U}$ are the feasible state and input sets. The optimization problem is solved at each control time step in a receding-horizon fashion.

Rather than combining all tasks into a single weighted objective, lexicographic optimization preserves strict priority among tasks. In practice, this is commonly implemented by solving a sequence of single-task MPC (STMPC) problems from the highest-priority task to the lowest-priority task. At iteration l , the problem can be written as

$$\mathbf{x}_k^{l*}(\cdot), \mathbf{u}_k^{l*}(\cdot) = \arg \min_{\mathbf{x}(\cdot), \mathbf{u}(\cdot)} \mathcal{J}_l + \mathcal{J}_{aux} \quad (12a)$$

$$\text{s.t. (11b) – (11e)}, \quad (12b)$$

$$\mathcal{J}_i(\mathbf{x}(\cdot), \mathbf{u}(\cdot)) \leq \mathcal{J}_i(\mathbf{x}_k^{i*}(\cdot), \mathbf{u}_k^{i*}(\cdot)), \quad (12c)$$

$$\forall i = 1, \dots, l-1$$

where $(\mathbf{x}_k^{l*}(\cdot), \mathbf{u}_k^{l*}(\cdot))$ is the state-input trajectory obtained for task τ_l over the prediction time interval $\mathbb{T}_k$, and $\mathcal{J}_{aux}$ is an auxiliary cost used to improve control performance and numerical conditioning. The task-cost constraints in (12) prevent later optimizations from increasing the costs attained for previously solved higher-priority tasks. Consequently, only the remaining redundancy after satisfying the higher-priority task-cost bounds is used to realize lower-priority behaviors. HTMPC has been used to enforce the temporal order in sequential mobile manipulation tasks [16], [17]. In this paper, we use HTMPC to track the desired base motion as the primary task while using residual redundancy to realize perception-oriented hand camera motions. In the following sections, we

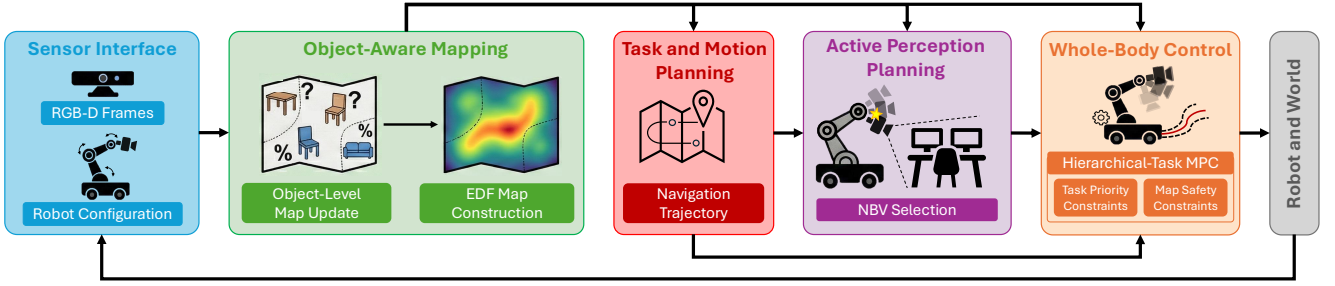


Fig. 3: An overview of the proposed active perception and navigation framework for high-DoF mobile manipulators.

drop the time dependence in state and control inputs wherever it is clear.

IV. SYSTEM DESCRIPTION

A. System Overview

We present an active perception and navigation framework for mobile manipulators operating in cluttered, unknown, and semi-static environments. Our robot platform, shown in Fig. 4, uses two synchronized RGB-D cameras: a *base camera* rigidly mounted on the mobile base and a *hand camera* mounted on the manipulator end effector. The robot is assigned a sparse set of navigation waypoints as its primary task, while the hand camera is allowed to reorient online to inspect uncertain, occluded, or safety-critical regions. The complete closed-loop pipeline is illustrated in Fig. 3.

The system contains five modules: a sensor interface, an object-aware perception and mapping module, a path-planning module, an object- and task-aware NBV planner, and a planning and control module. At each time step t_k , camera, odometry, and joint measurements are synchronized to estimate the robot state, the map is updated, and a desired base path is planned from the current map and given waypoints. The active perception module then evaluates candidate viewpoints using their predicted reduction in task-dependent map entropy. The control module tracks the base path while realizing the selected

hand camera objective, producing new observations that close the perception-action loop.

B. Robot State and State Estimation

We first define the robot state used throughout the framework. Let $\mathbf{q} \in \mathbb{R}^n$ denote the generalized coordinate, $\mathbf{v} \in \mathbb{R}^m$ the generalized velocity, and $\mathbf{u} = \dot{\mathbf{v}} \in \mathbb{R}^m$ the acceleration input. The full state is $\mathbf{x} = [\mathbf{q}^T, \mathbf{v}^T]^T$, with feasible state and input sets $\mathcal{X}$ and $\mathcal{U}$ respectively. The state is partitioned into mobile-base and arm components using subscripts b and a . For our platform, $\mathbf{q}_{b,k}, \mathbf{v}_{b,k} \in \mathbb{R}^3$ describe the Ridgeback planar base pose and velocity, while $\mathbf{q}_{a,k}, \mathbf{v}_{a,k} \in \mathbb{R}^6$ describe the UR10 joint positions and velocities, resulting in $n = m = 9$. At discrete time t_k , the robot state is written as

$$\mathbf{x}_k = [\mathbf{q}_{b,k}^T \quad \mathbf{q}_{a,k}^T \quad \mathbf{v}_{b,k}^T \quad \mathbf{v}_{a,k}^T]^T. \quad (13)$$

The sensor interface synchronizes camera, odometry, and joint measurements to produce an estimated robot state $\hat{\mathbf{x}}_k$. At time t_k , let $\mathbf{F}_k^{\text{base}}$ and $\mathbf{F}_k^{\text{hand}}$ denote the RGB-D frames captured by the base and hand cameras, $\mathbf{y}_k^{\text{odo}}$ the wheel odometry measurement, and $\mathbf{q}_{a,k}, \mathbf{v}_{a,k}$ the arm encoder measurements. Together with the known camera extrinsics, these measurements form the estimator input.

The base pose and velocity $\mathbf{q}_{b,k}$ and $\mathbf{v}_{b,k}$ are estimated using a modified ORB-SLAM3 [18] with the base camera and wheel odometry. Wheel odometry improves robustness in feature-sparse regions and interpolates or extrapolates the base state to compensate for RGB-D frame-rate and latency limits. In parallel, the current hand camera pose is estimated using the measured arm joint states $\mathbf{q}_{a,k}$ and $\mathbf{v}_{a,k}$ through forward kinematics. The resulting state estimate $\hat{\mathbf{x}}_k$, together with the hand camera pose, is passed to the mapping, NBV, and control modules as their common robot state input. For the rest of the paper, we omit the hat and let $\mathbf{x}_k$ denote the estimated robot state for readability.

C. Object-Aware Perception and Map Maintenance

We expect the robot to operate in semi-static environments where rigid objects may appear, disappear, or shift over time. When deployed in a new environment, the robot starts with an empty map and incrementally updates it as it traverses the scene. To maintain an up-to-date scene representation, the RGB-D frames from both cameras are unprojected into the global frame, segmented into point-cloud observations $\mathcal{Y}_k = \{\mathbf{Y}_{k,p}\}_{p=1 \dots P}$ via geometric clustering, and merged into the mapped object library $\mathcal{O}$ through optimal association.

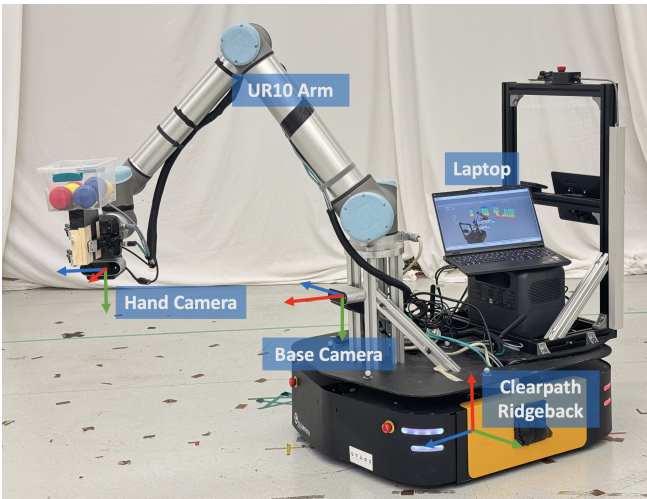


Fig. 4: Our mobile manipulator setup. The system consists of a Universal Robots UR10 arm mounted on a Clearpath Robotics Ridgeback omnidirectional base. Two synchronized Orbbec Femto Bolt RGB-D cameras are installed, with one mounted on the base and the other mounted on the hand. An onboard laptop runs all the perception, planning and control algorithms.

This module forms the map-maintenance core of the framework. The object representation and probabilistic state update follow the semi-static mapping model introduced in Section III-A. New observations spawn new objects, while matched observations update existing objects and propagate their state distributions through Bayesian inference. If an object's consistency expectation in (2) falls below a threshold, it is treated as changed and reconstructed in the maintained map. In addition to maintaining object-level consistency, this module produces the uncertainty estimates used in the downstream NBV module. Near mapped objects, the map uncertainty is inherited from the corresponding object consistency variance in (3). In non-object regions, it is represented by a Beta occupancy uncertainty model. If an object is not observed for too long, its state model is additionally penalized by synthetic outlier evidence so that outdated objects become increasingly uncertain and are encouraged to be re-observed. The construction of these uncertainty fields is detailed in Section V-A.

D. Object- and Task-Aware Next-Best-View Planning

The active perception module receives the maintained map, estimated robot state, and desired base path, and selects a hand camera viewpoint predicted to reduce task-dependent map entropy. As illustrated in Fig. 5, it ray traces the cells visible from each of R sampled poses, weights them using the predicted base motion along the path, and scores each pose using the approximate EIG defined in Section V-F.

Object-level consistency identifies potentially changed regions, while path-dependent weighting prioritizes those relevant to navigation. Together, these components favor observations that can update the map before an upcoming maneuver.

The maintained map incorporates the hand-camera view history $\mathcal{T}_k$. Given R candidate poses $\{\mathbf{T}_{k+1,r}\}_{r=1}^R$, the selected NBV is

$$\mathbf{T}_{k+1}^d = \arg \max_r \text{EIG}(\mathbf{T}_{k+1,r}), \quad (14)$$

where $\text{EIG}(\cdot)$ is the approximate expected information-gain score. The selected pose becomes the hand-camera reference for control. The map uncertainty, path weighting, and viewpoint score are developed in Section V-A–Section V-F.

E. Whole-Body Control

The whole-body controller receives the desired base path, current robot and map state, and selected hand-camera reference. It coordinates base tracking and camera motion using the ordered task list

$$[\tau_{\text{base}}, \tau_{\text{cam}}], \quad (15)$$

where τ_{base} tracks the base path and τ_{cam} targets the NBV pose. HTMPC gives higher priority to base tracking, uses remaining redundancy for camera motion, and enforces map-derived collision constraints. Both references are updated as the map changes. The controller is formulated in Section VI.

V. NEXT-BEST-VIEW FORMULATION

This section formalizes the proposed object- and task-aware NBV module. We first construct the maintained map uncertainty and desired-path prior, then derive the task-dependent

map entropy, predict its update for each candidate view, and select the NBV using approximate expected information gain.

A. Map and Uncertainty Construction

The mapping module maintains an object library $\mathcal{O} = \{\mathbf{O}_i\}$ where each object $\mathbf{O}_i$ carries a local EDF submap $\mathcal{M}_i$ and a state model $p(l_i, v_i)$. We merge these submaps into a single 3D EDF used in whole-body control for collision avoidance by taking the minimum EDF value over all object submaps,

$$\mathcal{M}_{\text{edf}}(\cdot) = \min_{\mathbf{O}_i \in \mathcal{O}} \mathcal{M}_i(\cdot). \quad (16)$$

The resulting $\mathcal{M}_{\text{edf}}$ is the 3D collision map later interpolated in Section VI for whole-body collision avoidance [17].

For navigation and NBV, we evaluate this map on a 2D grid $\mathcal{G} = \{(m, n)\}$ defined at the base height z_b , with cell centers $\mathbf{c}_{mn} = [x_{mn}, y_{mn}]^T$. The sliced EDF value at each cell is

$$u_s(\mathbf{c}_{mn}) := \mathcal{M}_{\text{edf}}(x_{mn}, y_{mn}, z_b), \quad (m, n) \in \mathcal{G}. \quad (17)$$

Bilinear interpolation over this discrete slice then yields the continuous approximation $u_s(\mathbf{p})$ used in the NBV Fisher information derivation. For each slice cell, we define the index of the closest mapped object by

$$i_{mn}^* := \arg \min_{\mathbf{O}_i \in \mathcal{O}} \mathcal{M}_i(x_{mn}, y_{mn}, z_b). \quad (18)$$

A cell is treated as object-associated when its closest mapped object lies within a distance band δ_o , i.e., when $u_s(\mathbf{c}_{mn}) \leq \delta_o$. For these cells, the map uncertainty is inherited from the corresponding object-consistency variance $\text{Var}[v_{i_{mn}^*}]$ in (3). If an object is not re-observed for an extended period, the mapping module gradually penalizes it with synthetic outlier measurements, which lowers the consistency estimate and enlarges this variance without forcing immediate object removal.

The shared object belief couples the uncertainty assigned to its associated cells. For example, observations of a shifted cabinet's visible edge can change the maintained consistency belief for that cabinet, affecting the uncertainty of its associated region even where the surface remains occluded. This gives subsequent viewpoint selection a reason to inspect the rest of the object. Path weighting, introduced next, determines how strongly that region matters to the navigation task.

Cells with $u_s(\mathbf{c}_{mn}) > \delta_o$ are modeled independently by a Beta occupancy model,

$$o_{mn,k} \sim \text{Beta}(\alpha_{mn,k}^{\text{occ}}, \beta_{mn,k}^{\text{occ}}), \quad (19)$$

whose parameters are updated from a binary occupancy observation $z_{mn,k}^{\text{occ}} \in \{0, 1\}$ whenever the cell is visible:

$$\begin{aligned} \alpha_{mn,k+1}^{\text{occ}} &= \alpha_{mn,k}^{\text{occ}} + \eta_{\text{occ}} z_{mn,k}^{\text{occ}}, \\ \beta_{mn,k+1}^{\text{occ}} &= \beta_{mn,k}^{\text{occ}} + \eta_{\text{occ}} (1 - z_{mn,k}^{\text{occ}}), \end{aligned} \quad (20)$$

where $\eta_{\text{occ}} > 0$ is an observation weight and $\eta_{\text{occ}} = 1$ recovers the standard Beta-Bernoulli update. If a non-object cell is not re-observed, its parameters are relaxed toward an uninformative prior $(\alpha_0^{\text{occ}}, \beta_0^{\text{occ}})$ to reflect growing uncertainty:

$$\begin{aligned} \alpha_{mn,k+1}^{\text{occ}} &= (1 - \rho) \alpha_{mn,k}^{\text{occ}} + \rho \alpha_0^{\text{occ}}, \\ \beta_{mn,k+1}^{\text{occ}} &= (1 - \rho) \beta_{mn,k}^{\text{occ}} + \rho \beta_0^{\text{occ}}, \end{aligned} \quad (21)$$

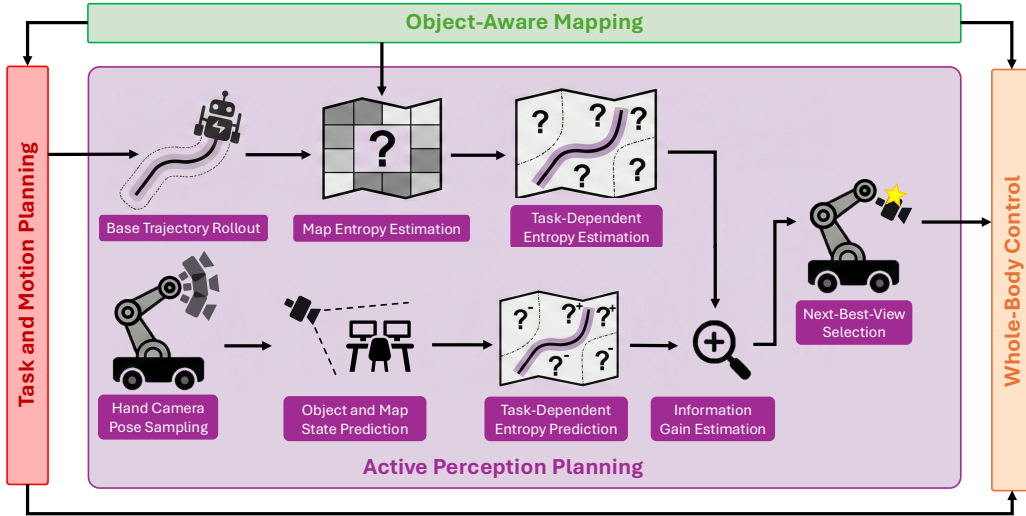


Fig. 5: Detailed schematic of the proposed active perception module.

where $\rho \in (0, 1)$ is a forgetting factor. Since $o_{mn,k}$ is also Beta distributed, its variance follows the same form as (3), with $(\alpha_{i,k}, \beta_{i,k})$ replaced by $(\alpha_{mn,k}^{\text{occ}}, \beta_{mn,k}^{\text{occ}})$.

Finally, we define the effective cell-wise EDF variance used in the NBV objective as

$$\sigma_{s,mn}^2 := \begin{cases} \lambda_{\text{edf}} \text{Var}[v_{i_{mn}^*}], & u_s(\mathbf{c}_{mn}) \leq \delta_o, \\ \lambda_{\text{edf}} \text{Var}[o_{mn,k}], & \text{otherwise,} \end{cases} \quad (22)$$

where $\lambda_{\text{edf}} > 0$ is a scaling factor. Note that $\text{Var}[v_i]$ in (3) and $\text{Var}[o_{mn,k}]$ describe object consistency and map occupancy uncertainties, not EDF uncertainty, $\text{Var}[\text{EDF}(\mathbf{c}_{mn})]$, directly. Prior work in uncertainty-aware object mapping derives a nonlinear forward propagation from latent occupancy uncertainty to signed-distance variance [57]. However, the inverse mapping needed here is not available in closed form. We therefore use the latent consistency and occupancy variances as monotone surrogates and interpret (22) as an effective linear EDF-uncertainty model. The merged 3D map provides the distance field for whole-body collision avoidance, while the 2D slice provides the distance field for base path planning together with the uncertainty field used for NBV selection.

B. Desired Path and Robot Position Prior

The navigation task is represented by a dense reference trajectory with length I :

$$\mathbf{X}^d = \{\mathbf{p}_i^d\}_{i=0,\dots,I}, \quad (23)$$

where $\mathbf{p}_i^d \in \mathbb{R}^2$ denotes the desired 2D base position at rollout step $t_i = i\Delta t$ along the reference path. During execution, the robot may deviate from this path because of tracking error, collision avoidance, and replanning. We therefore model the robot position as a stochastic rollout around the reference path. This rollout provides the spatial task-relevance weights for NBV, while HTMPC uses a separate, short prediction horizon for control.

For the holonomic base, let $\mathbf{p}_i = [p_{x,i} \ p_{y,i}]^T$ denote the 2D rollout position and $\bar{\mathbf{u}}_i = [v_{x,i} \ v_{y,i}]^T$ denote the nominal

velocity derived from the desired path produced by the global planner. We model the executed velocity as

$$\mathbf{u}_i = \bar{\mathbf{u}}_i + \mathbf{w}_i, \quad \mathbf{w}_i \sim \mathcal{N}(\mathbf{0}, \sigma_d^2 \mathbf{I}), \quad (24)$$

where $\mathbf{w}_i$ is zero-mean process noise with variance $\sigma_d^2 \mathbf{I}$ that captures tracking deviations and local replanning effects. This yields the base position rollout

$$\mathbf{p}_{i+1} = \mathbf{p}_i + \Delta t \cdot \mathbf{u}_i. \quad (25)$$

Under this random-walk approximation, the position covariance evolves as

$$\Sigma_{d,i} = \Sigma_{d,0} + i \cdot (\Delta t^2 \sigma_d^2 \mathbf{I}). \quad (26)$$

Accordingly, we discretize a Gaussian position model centered at $\mathbf{p}_i^d$ with covariance $\Sigma_{d,i}$ over the grid and normalize the resulting cell weights:

$$\begin{aligned} \tilde{w}_{mn|i} &:= \exp\left(-\frac{1}{2}(\mathbf{c}_{mn} - \mathbf{p}_i^d)^T \Sigma_{d,i}^{-1}(\mathbf{c}_{mn} - \mathbf{p}_i^d)\right), \\ p(\mathbf{c}_{mn} | t_i) &:= \frac{\tilde{w}_{mn|i}}{\sum_{(a,b) \in \mathcal{G}} \tilde{w}_{ab|i}}. \end{aligned} \quad (27)$$

We use a uniform prior $p(t_i) = 1/(I+1)$ over the rollout steps $i = 0, \dots, I$. This normalized path-conditioned distribution weights map regions by how likely the robot is to occupy them along the planned path, using the same grid $\mathcal{G}$ for every candidate view.

C. Task-Dependent Map Entropy

At time step t_k , the accumulated hand camera view history is denoted by

$$\mathcal{T}_k = \{\mathbf{T}_1, \dots, \mathbf{T}_k\}. \quad (28)$$

Conditioned on $\mathcal{T}_k$, the maintained 2D EDF slice and its uncertainty define the current map belief used by NBV. For a planar position $\mathbf{p}$ lying in cell (m, n) , using the sliced EDF mean in (17) and the effective EDF variance in (22), the local distance-to-nearest-obstacle observation is modeled as

$$\begin{aligned} \text{EDF}(\mathbf{p} | \mathcal{T}_k) &= u_s(\mathbf{p}) + \eta_{s,mn}, \\ \eta_{s,mn} &\sim \mathcal{N}(0, \sigma_{s,mn}^2), \end{aligned} \quad (29)$$

where $u_s(\mathbf{p})$ is the interpolated EDF mean and $\sigma_{s,mn}^2$ is the cell uncertainty in (22). The term $\eta_{s,mn}$ denotes the corresponding local Gaussian EDF noise, and the map gradient $\nabla u_s(\mathbf{p})$ is estimated by finite differences over the grid.

In our formulation, the unknown parameter is a potential robot base position $\mathbf{p}$ and the observation is the EDF value at that position. Since we approximate the EDF observation as Gaussian, the obstacle-distance likelihood for a robot base position $\mathbf{p}$ inside cell (m, n) is

$$p(\text{EDF} \mid \mathcal{T}_k, \mathbf{p}) = p(\text{EDF} \mid u_s(\mathbf{p}), \sigma_{s,mn}^2) \propto \exp\left(-\frac{(\text{EDF} - u_s(\mathbf{p}))^2}{2\sigma_{s,mn}^2}\right). \quad (30)$$

Ignoring constant terms, the corresponding log-likelihood is

$$\log p(\text{EDF} \mid \mathcal{T}_k, \mathbf{p}) \sim -\frac{1}{2}\sigma_{s,mn}^{-2}(\text{EDF} - u_s(\mathbf{p}))^2. \quad (31)$$

The Fisher score then becomes

$$\nabla \log p(\text{EDF} \mid \mathcal{T}_k, \mathbf{p}) = \sigma_{s,mn}^{-2}(\text{EDF} - u_s(\mathbf{p}))\nabla u_s(\mathbf{p}). \quad (32)$$

Evaluating the FIM at the center of cell (m, n) yields

$$\begin{aligned} \mathcal{F}(\mathbf{c}_{mn} \mid \mathcal{T}_k) &= \mathbb{E}_{\text{EDF}} \left[(\nabla \log p(\text{EDF} \mid \mathcal{T}_k, \mathbf{c}_{mn})) \right. \\ &\quad \left. \cdot \nabla \log p(\text{EDF} \mid \mathcal{T}_k, \mathbf{c}_{mn})^T \right] \\ &= \sigma_{s,mn}^{-2} \nabla u_s(\mathbf{c}_{mn}) \nabla u_s(\mathbf{c}_{mn})^T. \end{aligned} \quad (33) \quad (34)$$

Cells with zero EDF gradient are omitted from entropy evaluation. For the remaining cells, $\mathcal{F}(\mathbf{c}_{mn} \mid \mathcal{T}_k)$ is rank-1 and therefore non-invertible. Intuitively, the robot only gains information about the closest obstacle along the local map-gradient direction $\nabla u_s(\mathbf{c}_{mn})$. Its non-zero eigenvalue is

$$\lambda_{mn} = \sigma_{s,mn}^{-2} \|\nabla u_s(\mathbf{c}_{mn})\|^2. \quad (35)$$

Under a local linearization, λ_{mn}^{-1} gives the variance scale along the EDF-gradient direction. The differential entropy of the corresponding 1D Gaussian approximation is

$$h_e^{1D}(\mathbf{c}_{mn}; \mathcal{T}_k) = \frac{1}{2} \log \left[2\pi e \cdot \frac{\sigma_{s,mn}^2}{\|\nabla u_s(\mathbf{c}_{mn})\|^2} \right]. \quad (36)$$

From this expression, the 1D entropy along the map-gradient direction increases with larger cell uncertainty $\sigma_{s,mn}^2$ and smaller gradient magnitude $\|\nabla u_s(\mathbf{c}_{mn})\|$. Intuitively, a cell is more uncertain either because its state is less certain or because the local EDF varies weakly with position, making a distance observation less informative. In free-space regions with $u_s(\mathbf{c}_{mn}) > \delta_o$, using the same object-association threshold δ_o introduced in Section V-A, we further clip the EDF gradient magnitudes from above to avoid overconfidence in previously observed empty space. This increases the entropy assigned to such regions and encourages the robot to re-observe them for newly added obstacles. At each grid cell, we compute this one-dimensional Fisher-derived entropy from the effective EDF uncertainty and local map gradient. We define task-dependent map entropy as its expectation over the normalized

path-conditioned distribution of grid cells and rollout times:

$$\begin{aligned} \mathbb{E}_{\mathbf{p},t}[h_e^{1D}(\mathcal{T}_k)] \\ = \sum_{i=0}^I \sum_{(m,n) \in \mathcal{G}} h_e^{1D}(\mathbf{c}_{mn}; \mathcal{T}_k) p(\mathbf{c}_{mn} \mid t_i) p(t_i). \end{aligned} \quad (37)$$

This task-dependent map entropy serves as the prior term for scoring each candidate viewpoint by approximate EIG.

D. Candidate Views and Visible-Cell Set

At time step t_k , let $\mathcal{V}_k \subset \text{SE}(3)$ denote the candidate hand-camera target pose space. The formulation is continuous in principle but we sample a finite candidate set from $\mathcal{V}_k$:

$$\{\mathbf{T}_{k+1,r}\}_{r=1,\dots,R} \subset \mathcal{V}_k, \quad (38)$$

where each $\mathbf{T}_{k+1,r}$ is a candidate hand-camera target for the next observation. Our path-guided implementation is described in Section VII. In the controller, camera orientation is enforced more strongly than position, allowing local position adjustment near obstacles while preserving the intended viewing direction.

Each candidate viewpoint captures a subset of grid cells within its field of view. We denote this visible-cell set by

$$\mathcal{C}_{k+1,r} := \mathcal{C}(\mathbf{T}_{k+1,r}), \quad (39)$$

which is obtained by ray tracing on the current maintained map to collect the grid cells expected to be visible from viewpoint $\mathbf{T}_{k+1,r}$. Thus, although the prior entropy in Section V-C is defined over the full grid, only cells in $\mathcal{C}_{k+1,r}$ undergo the predicted map update.

E. Predicted Map Update Under a Candidate View

Given a candidate viewpoint $\mathbf{T}_{k+1,r}$, we only predict the posterior for the visible cells in $\mathcal{C}_{k+1,r}$. Since multiple visible cells may belong to the same object, we first collect the associated object set induced by the visible object cells,

$$\mathcal{O}_{k+1,r} := \{\mathbf{O}_{i_{mn}^*} \mid \mathbf{c}_{mn} \in \mathcal{C}_{k+1,r}, u_s(\mathbf{c}_{mn}) \leq \delta_o\}. \quad (40)$$

For each object $\mathbf{O}_i \in \mathcal{O}_{k+1,r}$, we synthesize two geometric measurements: an inlier Δ_i^{in} to simulate that the object is stationary, and an outlier Δ_i^{out} to simulate that the object has changed, with $|\Delta_i^{\text{in}}| \ll |\Delta_i^{\text{out}}|$. Applying the Bayesian update in (6) with these two synthetic measurements yields two candidate posterior consistency variances, denoted by $\widehat{\text{Var}}^{\text{in}}[v_i]$ and $\widehat{\text{Var}}^{\text{out}}[v_i]$. We retain the posterior with the smaller variance and denote it by

$$\widehat{v}_i^{\text{obj}} := \min\left(\widehat{\text{Var}}^{\text{in}}[v_i], \widehat{\text{Var}}^{\text{out}}[v_i]\right). \quad (41)$$

This is an optimistic posterior approximation: rather than integrating over all possible future measurements, we evaluate two representative outcomes and use the lower posterior variance as an estimate of the uncertainty reduction achievable by observing the object. All visible cells associated with object $\mathbf{O}_i$ inherit this predicted variance.

Visible non-object cells with $u_s(\mathbf{c}_{mn}) > \delta_o$ are then updated independently. For each such cell, we synthesize two binary occupancy measurements, occupied and free, apply the

weighted Beta-Bernoulli update in (20), and denote the corresponding posterior variances by $\widehat{\text{Var}}^1[o_{mn,k}]$ and $\widehat{\text{Var}}^0[o_{mn,k}]$. We use the same approximation and retain the smaller one:

$$\widehat{V}_{mn}^{\text{occ}} := \min\left(\widehat{\text{Var}}^1[o_{mn,k}], \widehat{\text{Var}}^0[o_{mn,k}]\right). \quad (42)$$

For compactness, denote the predicted effective EDF variance under candidate r as $\widehat{\sigma}_{s,mn,r}^2$:

$$\widehat{\sigma}_{s,mn,r}^2 := \begin{cases} \lambda_{\text{edf}} \widehat{V}_{mn}^{\text{obj}}, & \mathbf{c}_{mn} \in \mathcal{C}_{k+1,r}, u_s(\mathbf{c}_{mn}) \leq \delta_o, \\ \lambda_{\text{edf}} \widehat{V}_{mn}^{\text{occ}}, & \mathbf{c}_{mn} \in \mathcal{C}_{k+1,r}, u_s(\mathbf{c}_{mn}) > \delta_o, \\ \sigma_{s,mn}^2, & \mathbf{c}_{mn} \notin \mathcal{C}_{k+1,r}. \end{cases} \quad (43)$$

To predict candidate-dependent geometry, we mark a thin layer of currently free cells along the candidate FoV frontier as hypothetically occupied and recompute the EDF $\widehat{u}_{s,r}$ and its gradient within $\mathcal{C}_{k+1,r}$. This geometric hypothesis is applied independently of which synthetic observation yields the smaller variance. Outside the FoV, we set $\widehat{u}_{s,r} = u_s$, neglecting induced changes there for efficiency. The predicted posterior entropy used in the EIG calculation is

$$h_e^{1D}(\mathbf{c}_{mn}; \mathcal{T}_k, \mathbf{T}_{k+1,r}) = \frac{1}{2} \log \left[2\pi e \cdot \frac{\widehat{\sigma}_{s,mn,r}^2}{\|\nabla \widehat{u}_{s,r}(\mathbf{c}_{mn})\|^2} \right]. \quad (44)$$

F. Expected Information Gain and NBV Selection

Section V-C derives the current task-dependent map entropy, and Section V-E derives its predicted value under each candidate viewpoint. After incorporating candidate viewpoint $\mathbf{T}_{k+1,r}$, the task-dependent map entropy is

$$\begin{aligned} & \mathbb{E}_{\mathbf{p},t}[h_e^{1D}(\mathcal{T}_k, \mathbf{T}_{k+1,r})] \\ &= \sum_{i=0}^I \sum_{(m,n) \in \mathcal{G}} h_e^{1D}(\mathbf{c}_{mn}; \mathcal{T}_k, \mathbf{T}_{k+1,r}) p(\mathbf{c}_{mn} | t_i) p(t_i). \end{aligned} \quad (45)$$

The approximate expected information gain (EIG) of candidate viewpoint $\mathbf{T}_{k+1,r}$ is then defined as the reduction from prior to predicted posterior task-dependent map entropy:

$$\begin{aligned} \text{EIG}(\mathbf{T}_{k+1,r}) &= \mathbb{E}_{\mathbf{p},t}[h_e^{1D}(\mathcal{T}_k)] \\ &\quad - \mathbb{E}_{\mathbf{p},t}[h_e^{1D}(\mathcal{T}_k, \mathbf{T}_{k+1,r})]. \end{aligned} \quad (46)$$

The expectation uses the normalized grid-cell and rollout-time distribution, with the optimistic posterior prediction defined in Section V-E. A larger approximate EIG therefore predicts a greater reduction in task-dependent map entropy. Under the FoV-local approximation, non-visible cells retain their prior map state, so their contributions cancel and only visible cells need to be considered:

$$\begin{aligned} \text{EIG}(\mathbf{T}_{k+1,r}) &= \sum_{i=0}^I \sum_{(m,n) \in \mathcal{C}_{k+1,r}} \left(h_e^{1D}(\mathbf{c}_{mn}; \mathcal{T}_k) \right. \\ &\quad \left. - h_e^{1D}(\mathbf{c}_{mn}; \mathcal{T}_k, \mathbf{T}_{k+1,r}) \right) p(\mathbf{c}_{mn} | t_i) p(t_i). \end{aligned} \quad (47)$$

Among the candidate set defined in Section V-D, the next-best viewpoint is finally selected as

$$\mathbf{T}_{k+1}^d = \arg \max_r \text{EIG}(\mathbf{T}_{k+1,r}). \quad (48)$$

VI. HIERARCHICAL-TASK MPC

The mobile manipulator assigns higher priority to tracking the desired base trajectory than to realizing the NBV hand camera poses. We employ HTMPC to enforce this task ordering and generate smooth whole-body motion for navigation while supporting active perception. The NBV module provides the selected hand-camera pose $\mathbf{T}_{k+1}^d$, which is held constant as the camera reference for a fixed update interval.

We consider a mobile manipulator with motion model

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{q})\mathbf{x} + \mathbf{B}\mathbf{u}, \quad (49)$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{n \times n} & \mathbf{G}(\mathbf{q}) \\ \mathbf{0}_{m \times n} & \mathbf{0}_{m \times m} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{0}_{n \times m} \\ \mathbf{I}_{m \times m} \end{bmatrix}. \quad (50)$$

The generalized coordinate $\mathbf{q}$ and velocity $\mathbf{v}$ are related by $\dot{\mathbf{q}} = \mathbf{G}(\mathbf{q})\mathbf{v}$, where $\mathbf{G}(\mathbf{q})$ is a block diagonal matrix with two components: an identity matrix for the arm and a position-dependent matrix for the mobile base [58]. We choose accelerations as the control inputs, $\mathbf{u} = \dot{\mathbf{v}}$. If the base is holonomic, the matrix $\mathbf{G}(\mathbf{q})$ degenerates to an identity matrix $\mathbf{G} = \mathbf{I}_{n \times n}$. As a result, the holonomic mobile manipulator can be conveniently modeled as a double integrator.

Given the ordered task list $[\tau_{\text{base}}, \tau_{\text{cam}}]$, HTMPC assigns higher priority to $\mathcal{J}_{\text{base}}$ than to $\mathcal{J}_{\text{cam}}$. Both tasks are formulated as pose-tracking tasks whose cost is the accumulated tracking error $\mathbf{e}$ over the prediction horizon $[t_k, t_k + T_{\text{mpc}}]$:

$$\mathcal{J} = \frac{1}{2} \int_{t_k}^{t_k + T_{\text{mpc}}} \|\mathbf{e}\|_{\mathbf{Q}_e}^2 dt, \quad (51)$$

where $\mathbf{Q}_e \in \mathbb{R}^{6 \times 6}$ is a positive-definite diagonal weighting matrix and can be used to trade off the position and orientation error. In practice, the $\mathbf{Q}_e$ matrix can be designed to penalize the hand camera orientation error more since small variations in view angle often produce more changes in visibility and information gain than small translations. The error vector $\mathbf{e}$ is given by

$$\mathbf{e} = \begin{bmatrix} \mathbf{e}_{\text{pos}} \\ \mathbf{e}_{\text{orn}} \end{bmatrix} = \begin{bmatrix} \mathbf{P}^d - \mathbf{P} \\ \text{Log}(\mathbf{R}^{-1}\mathbf{R}^d)^\vee \end{bmatrix} \quad (52)$$

with position error $\mathbf{e}_{\text{pos}}$ and orientation error $\mathbf{e}_{\text{orn}}$. The orientation error is computed using the matrix logarithm $\text{Log} : \text{SO}(3) \rightarrow \mathfrak{so}(3)$ followed by the vee operator $(\cdot)^\vee : \mathfrak{so}(3) \rightarrow \mathbb{R}^3$. The desired position $\mathbf{P}^d$ and orientation $\mathbf{R}^d$ come from the planned navigation path $\mathbf{X}^d$ for the base task and from the selected NBV hand-camera pose $\mathbf{T}_{k+1}^d$ for the camera task.

Environment and self-collision avoidance are explicitly incorporated as the nonlinear constraints in (11c). To approximate the robot's envelope, we cover the robot arm and base with a set of M collision spheres and use them as the collision model. Each collision sphere S_j has radius r_j and center $\mathbf{P}^{S_j}(\mathbf{q})$ obtained from forward kinematics.

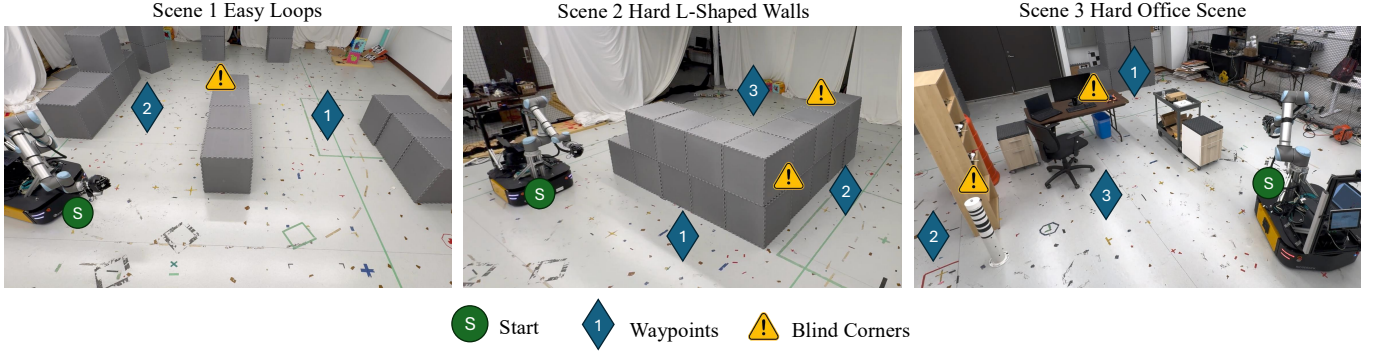


Fig. 6: Real-world static-scene benchmark. Left: Scene 1 Easy Loops. Middle: Scene 2 Hard L-Shaped Walls. Right: Scene 3 Hard Office Scene. The scenes progress from open loops to sharper turns and cluttered occlusions, where hidden hazards near the waypoints make active map maintenance important.

For environment collision avoidance, we impose a control-barrier-function (CBF)-based constraint for each collision sphere S_j . We define the collision-free safe set so that each sphere maintains a desired obstacle clearance δ_{safe} :

$$h_j(\mathbf{x}) = \text{Interp3D}(\mathcal{M}_{edf})(\mathbf{P}^{S_j}(\mathbf{q})) - r_j - \delta_{safe}, \quad (53)$$

$$\forall j = 1, 2, \dots, M,$$

where $\mathcal{M}_{edf}$ is the 3D EDF map constructed from the object library in Section V-A, and Interp3D denotes trilinear interpolation, making h_j continuous and piecewise differentiable. The CBF-based safety constraints are then

$$\dot{h}_j(\mathbf{x}, \mathbf{u}) > -\gamma h_j(\mathbf{x}), \quad \forall j = 1, 2, \dots, M. \quad (54)$$

Equation (54) is imposed as a clearance-rate constraint along the MPC trajectory. Intuitively, these constraints limit how quickly the robot can move toward obstacles. The constant γ acts as a decay factor that shapes this speed limit and the resulting traversal clearance. A larger γ allows the robot to behave more aggressively around obstacles, while a smaller γ enforces more conservative behavior. For self-collision avoidance, let $\mathcal{P}_{self}$ denote the selected set of nonadjacent collision-sphere pairs that can collide under the robot's joint limits. We impose a Euclidean-distance constraint for each pair:

$$\|\mathbf{P}^{S_i}(\mathbf{q}) - \mathbf{P}^{S_j}(\mathbf{q})\|_2 - r_i - r_j - \delta_{safe} \geq 0, \quad \forall (i, j) \in \mathcal{P}_{self}. \quad (55)$$

Both EDF- and CBF-based formulations are effective in idealized settings. In practice, however, the online constructed map is affected by perception delays, occlusions, and uncertainty, which are difficult to model explicitly in planning and control. We therefore choose the CBF formulation for environment collision avoidance and use the extended class function, through the parameter γ , to define more conservative behavior around obstacles [17], [53].

Collecting the cost functions and constraints defined above, the ideal lexicographic HTMPC problem is written as

$$\begin{aligned} \text{lex min}_{\mathbf{x}, \mathbf{u}} \quad & [\mathcal{J}_{base}, \mathcal{J}_{cam}] \\ \text{s.t.} \quad & (49), (54), (55), (11d), (11e). \end{aligned} \quad (56)$$

The base and camera objectives are minimized in lexicographic order subject to the motion model (49), safety constraints (54) and (55), state and input constraints (11d), and initial conditions (11e). The HTMPC controller is solved in a

receding-horizon fashion with the latest robot state, maintained map, planned base reference, and hand camera reference. In practice, we approximate the ideal hierarchy with two consecutive regularized STMPCs, preserving the base-tracking cost attained in the first stage. The first STMPC computes a numerically regularized base-tracking solution:

$$\begin{aligned} (\mathbf{x}_{base}^*, \mathbf{u}_{base}^*) = \arg \min_{\mathbf{x}, \mathbf{u}} \quad & \mathcal{J}_{base} + \mathcal{J}_{aux} \\ \text{s.t.} \quad & (49), (54), (55), (11d), (11e). \end{aligned} \quad (57)$$

The second STMPC optimizes the hand-camera task subject to an additional task-priority inequality that bounds the base-tracking cost by the value attained in the first stage:

$$\begin{aligned} (\mathbf{x}^*, \mathbf{u}^*) = \arg \min_{\mathbf{x}, \mathbf{u}} \quad & \mathcal{J}_{cam} + \mathcal{J}_{aux} \\ \text{s.t.} \quad & (49), (54), (55), (11d), (11e) \\ & \mathcal{J}_{base}(\mathbf{x}, \mathbf{u}) \leq \mathcal{J}_{base}(\mathbf{x}_{base}^*, \mathbf{u}_{base}^*). \end{aligned} \quad (58)$$

As a result of this inequality constraint, the camera task is optimized only within solutions that satisfy the attained base-tracking cost bound, encouraging the robot to use its remaining DoFs for active perception.

An auxiliary cost $\mathcal{J}_{aux}$ is added to both STMPC iterations to improve tracking performance and smoothness by penalizing velocity tracking error and state and control inputs:

$$\mathcal{J}_{aux} = \frac{1}{2} \int_{t_k}^{t_k + T_{mpc}} \|\dot{\mathbf{e}}\|_{\mathbf{Q}_e}^2 + \|\mathbf{x}\|_{\mathbf{Q}_x}^2 + \|\mathbf{u}\|_{\mathbf{Q}_u}^2 dt, \quad (59)$$

where $\mathbf{Q}_e$, $\mathbf{Q}_x$, and $\mathbf{Q}_u$ are positive definite weighting matrices of appropriate dimensions.

VII. EXPERIMENTS

A. Experimental Setup

1) *Platform and System Configuration:* The mobile manipulator platform and overall software architecture are introduced in Section IV, and here we summarize the experiment-specific settings. The perception and mapping stack builds on ORB-SLAM3 [18] and POCD [4]. A modified ORB-SLAM3 uses the base camera and wheel odometry to estimate the robot's state at 50 Hz, while dense object mapping integrates both base and hand camera observations at 5–10 Hz in parallel threads. We use a uniform semantic label and set the scene-change prior to $s_i = 0$ for all objects, so every mapped

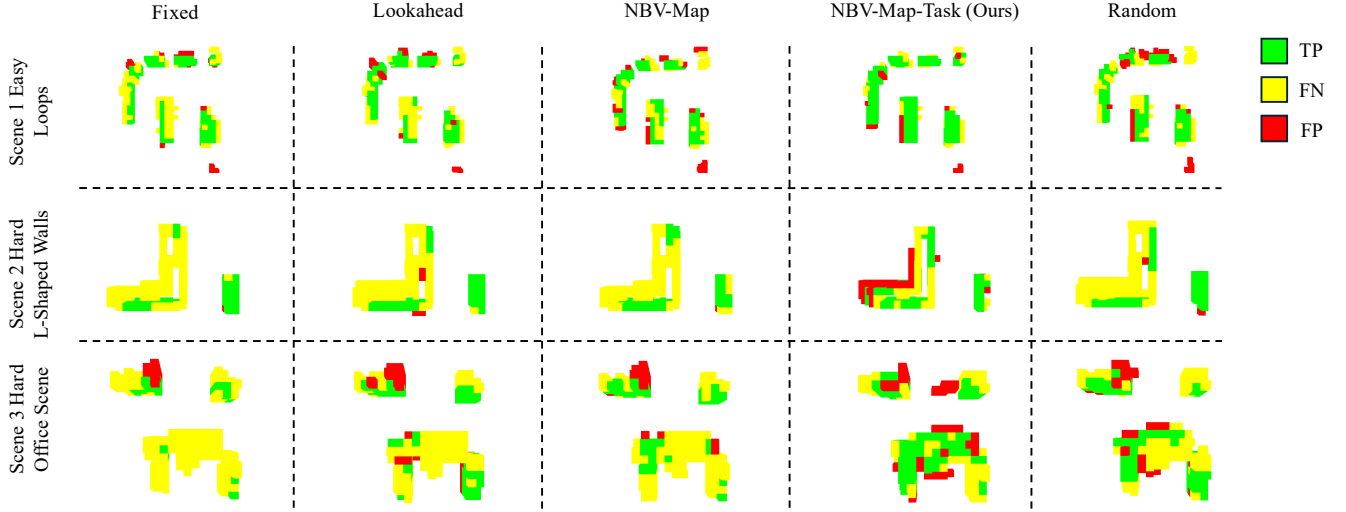


Fig. 7: Bird's-eye-view 3D reconstruction results in the real-world static-scene benchmark. Rows correspond to Scene 1–3. Green, yellow, and red indicate true positives, false negatives, and false positives, respectively. The proposed method recovers more occupied voxels in blind and path-critical regions.

TABLE I: Quantitative mapping results from the real-world static-scene benchmark. The signed improvement compares ours with the best baseline: ours minus baseline for precision and recall, and baseline minus ours for FPR; positive values favor ours.

Scene 1 Easy Loops	Precision $\uparrow$	Recall (TPR) $\uparrow$	FPR $\downarrow$	Outcome
Fixed	81.0	56.1	2.2	Success
Lookahead	81.7	59.6	2.3	Success
NBV-Map	81.8	61.9	2.2	Success
Random	77.2	56.2	2.8	Success
NBV-Map-Task (ours)	87.4	70.9	1.7	Success
<i>Signed improvement</i>	5.6	9.0	0.5	
Scene 2 Hard L-Shaped Walls	Precision $\uparrow$	Recall (TPR) $\uparrow$	FPR $\downarrow$	Outcome
Fixed	97.6	30.6	0.3	Collision $\times$ 3
Lookahead	89.9	31.2	1.6	Collision $\times$ 3
NBV-Map	97.4	28.4	0.3	Collision $\times$ 3
Random	94.0	30.4	0.9	Collision $\times$ 3
NBV-Map-Task (ours)	71.4	73.0	8.4	Success
<i>Signed improvement</i>	-26.2	41.8	-8.1	
Scene 3 Hard Office Scene	Precision $\uparrow$	Recall (TPR) $\uparrow$	FPR $\downarrow$	Outcome
Fixed	83.3	25.1	1.8	Collision $\times$ 3
Lookahead	73.1	47.5	4.4	Success
NBV-Map	80.0	37.9	3.4	Collision $\times$ 3
Random	78.7	58.3	5.4	Collision $\times$ 3
NBV-Map-Task (ours)	74.4	70.0	8.6	Success
<i>Signed improvement</i>	-8.9	11.7	-6.8	

object is treated as potentially changing. In the NBV module, $\lambda_{\text{edf}} = 10$, $\delta_o = 1.8$ m, and the EDF gradient magnitude is clipped to 0.8 in cells with $u_s(c_{mn}) > \delta_o$. Although the formulation allows sampling from the continuous reachable hand-camera pose space, our implementation uses a path-guided discrete set for runtime: a look-ahead waypoint defines ψ_k^{ref} , and we sample yaw offsets $\Delta\Psi = \{-\pi/3, -\pi/6, 0, \pi/6, \pi/3\}$ and we reaches $\mathcal{R} = \{1.5, 2.0\}$ m from the base center. Candidate reaches that fall too close to obstacles in the sliced EDF are shortened to preserve the safety margin. HTMPC is implemented in ACADOS [59] with $T_{\text{mpc}} = 3$ s, $\gamma = 0.7$, arm collision-sphere radius $r_j = 0.25$ m, and obstacle safety margin $\delta_{\text{safe}} = 0.15$ m. Both controllers run at 10 Hz, and the final commands are sent at 100 Hz. Unless otherwise noted, method parameters are fixed across experiments.

2) *Baseline Methods*: We compare the proposed object- and task-aware framework against four passive or partially informed baselines that differ only in how the hand camera pose is computed:

- *Fixed*: the hand camera is kept at a fixed position and points forward along the base's heading. This is a com-

mon default configuration for mobile manipulators and serves as a passive baseline that does not adapt to the scene or task [53].

- *Random*: the hand camera is selected uniformly at random from the same discrete candidate view set used by our method. This provides a fair active-perception baseline with the same motion range and view discretization, but without map or task awareness.
- *Lookahead*: the hand camera points toward a look-ahead point 8 s ahead along the desired path [17]. This baseline exploits a simple task-aware heuristic without reasoning about the maintained map state and uncertainty.
- *NBV-Map*: the hand camera is selected by the same information-theoretic NBV objective as our method, but without the task-weighting term. This baseline follows the more typical map exploration strategy of selecting views from maintained map uncertainty alone, without prioritizing regions relevant to navigation [9], [38].

We denote the full method by *NBV-Map-Task*. Together, the baselines separate passive sensing (Fixed), camera reorientation without map or task reasoning (Random), path-directed sensing (Lookahead), and uncertainty-driven sensing without task weighting (NBV-Map). The comparisons assess how combining map uncertainty and task relevance changes the observations acquired during navigation.

3) *Experiment Protocol*: We evaluate our framework across three experiment families. The static benchmark isolates view-point selection under occlusion. The controlled simulation then tests whether moved objects redirect sensing when the navigation task stays the same. Finally, repeated real-world visits examine whether those map updates support adaptive navigation under scene changes. Together, these experiments progress from acquiring missing geometry to revising previously acquired scene knowledge:

- **Real-world static-scene benchmark**: we conduct a benchmark in the three static, cluttered scenes shown in Fig. 6: Scene 1 Easy Loops, Scene 2 Hard L-Shaped Walls, and Scene 3 Hard Office Scene. Each method is

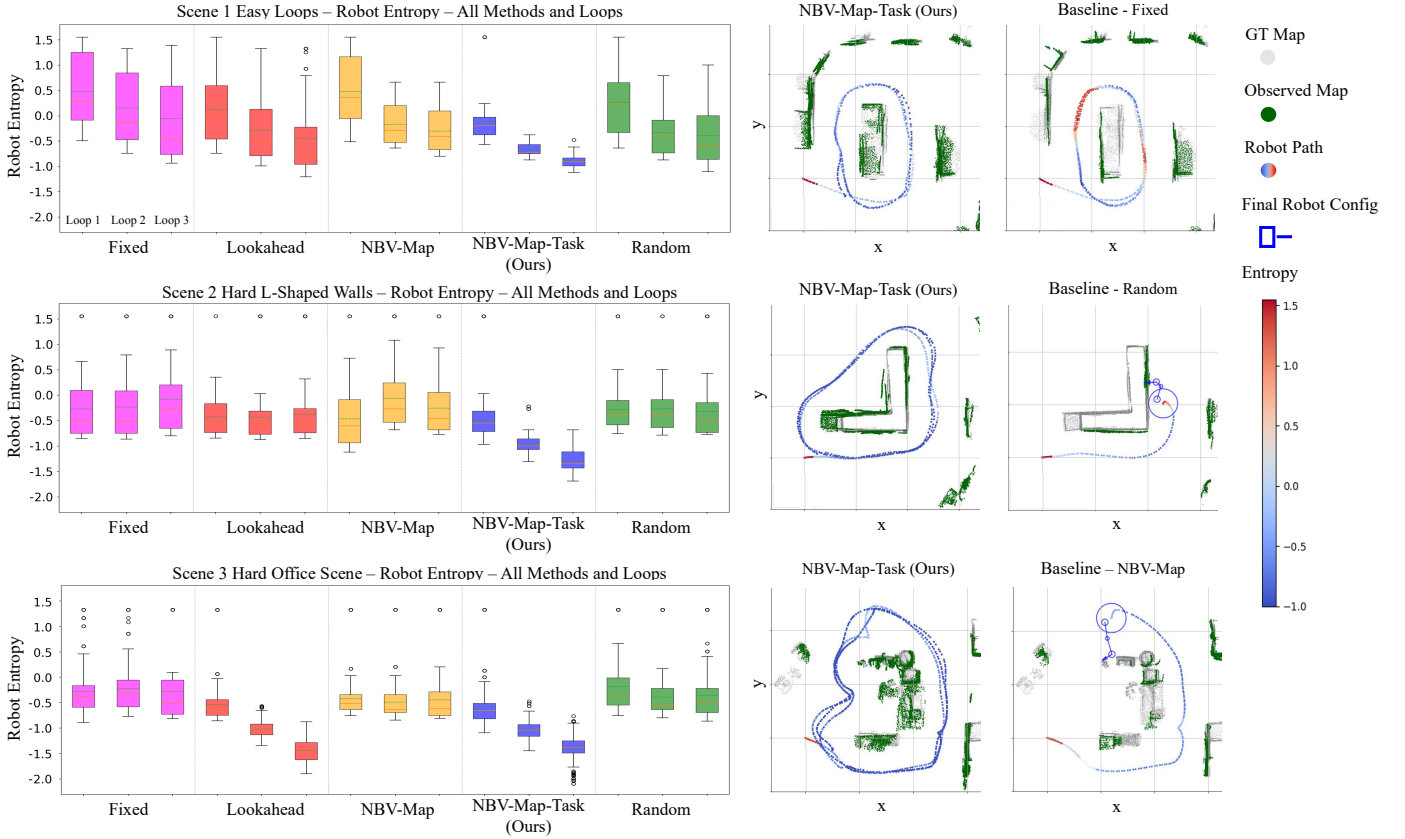


Fig. 8: Task-dependent map entropy along executed paths in the real-world static-scene benchmark. Left: boxplots over the three loops in each scene. Right: representative top-down views with trajectories colored by the local entropy. Lower values indicate lower entropy in the regions traversed by the robot.

executed for three consecutive loops in each scene to evaluate mapping quality, task-dependent map entropy, and navigation safety under repeated visits. The first loop starts with an empty map. If a loop succeeds without collision, the maintained map is carried over and updated in the next loop. Collisions trigger a robot and map reset.

- **Controlled scene-change simulation:** we use a controlled Isaac Sim simulation in which objects are shifted between loops to demonstrate how the object-aware uncertainty model qualitatively and quantitatively responds to object-level scene changes.
- **Long-term real-world scene-change experiment:** we perform a long-term real-world experiment in the office scene where a large shelf is relocated between visits to demonstrate incremental map maintenance and adaptive navigation behavior in a changing environment.

The multimedia attachment and shared video folder provide videos of the experiments.

B. Real-World Experiments in Static Scenes

1) *Scene Overview and Evaluation Metrics:* In this subsection, we evaluate the proposed system in the three real-world scenes shown in Fig. 6. This benchmark assesses whether the robot can reduce uncertainty and reliably detect obstacles in blind and path-critical regions before executing a turn or passing through a narrow space. The three scenes contain blind corners, narrow passages, and occluded obstacles along the planned path. With passive or task-unaware hand camera

motion, hidden obstacles may remain unobserved until the robot has already entered the risky region. In contrast, the proposed method directs the hand camera toward these regions earlier and incorporates the observations before the maneuver.

In all scenes, the robot receives only the blue waypoints shown in Fig. 6 and plans its path online from the maintained map. Yellow exclamation marks identify representative hidden hazards. Scene 1 Easy Loops provides an open setting, while Scene 2 Hard L-Shaped Walls and Scene 3 Hard Office Scene contain sharper turns and stronger occlusions. Starting without a map, the robot may initially plan through unseen obstacles. Successful execution therefore requires observing them before entering the turn.

We report this benchmark in three parts:

- **Map reconstruction quality:** precision, recall, and false positive rate on voxelized maps, after the final loop.
- **Task-dependent map entropy:** evaluated along the executed base trajectory and reported over all loops.
- **Navigation safety:** success rate and distance to the nearest reference-map obstacle, aggregated over all loops.

For each real-world scene, we construct a reference map by manually driving the robot to cover as much of the environment as possible. Both the maintained and reference 3D point clouds are voxelized at 0.3 m resolution. Evaluation is performed over the union of the fixed reference map and each method’s reconstructed map. Precision measures how many reconstructed occupied voxels agree with the reference map, recall (TPR) measures how many reference occupied

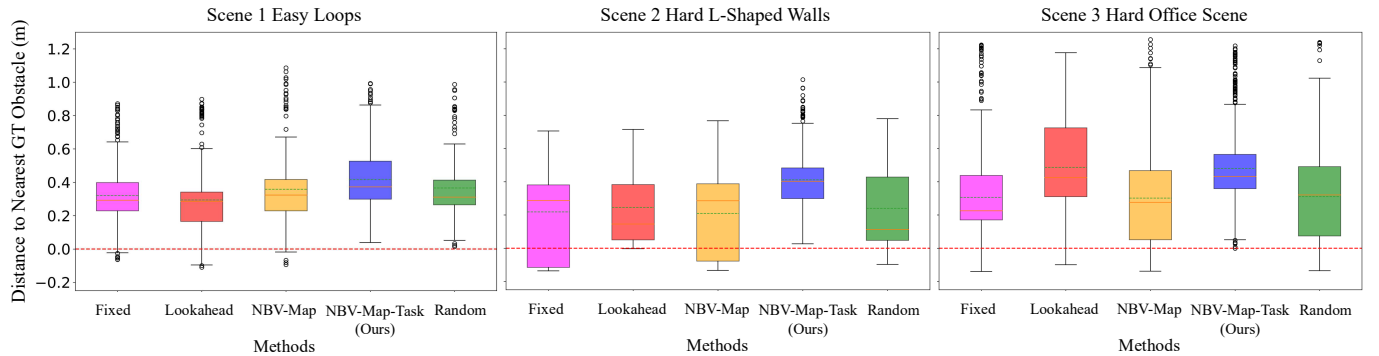


Fig. 9: Safety margin in the real-world static-scene benchmark. Boxplots show the point-cloud clearance proxy over all loop executions. The dashed red line marks zero clearance under this proxy, corresponding to a near-contact configuration. Larger positive values indicate greater estimated clearance, especially in the hard turning scenes.

TABLE II: Completion percentage across nine sequential loop executions per method: three loops in each of three scenes.

Method	Success Rate $\uparrow$
Fixed	33.3%
Lookahead	66.7%
NBV-Map	33.3%
Random	33.3%
NBV-Map-Task (ours)	100.0%

voxels are recovered, and FPR measures how many false occupied voxels appear relative to the reference map. High recall is especially important for navigation because missed occupied voxels can cause the robot to plan through unobserved obstacles. High precision and low FPR indicate fewer spurious occupied voxels, but are not sufficient if large obstacle regions remain unseen. Task-dependent map entropy is evaluated along the executed base trajectory in every loop. Clearance uses the same collision-sphere model as control: we evaluate the distance from each joint sphere surface to the nearest point in the reference point cloud and take the minimum over all joints. Reconstruction is evaluated after the final loop. Entropy, success, and clearance include all loop executions.

2) *Map Reconstruction Quality*: We first compare the voxel maps reconstructed at the end of the final traversal of each scene. This comparison assesses the reconstruction of occupied regions that determine path feasibility over repeated traversals, with map resets after collisions as described above. Fig. 7 shows the reconstructions, with voxel metrics in Table I.

In the easier Scene 1 setup, all methods complete the task, and the proposed method achieves the best precision, recall, and FPR. The recall gap becomes larger in Scene 2 and Scene 3, where critical obstacles lie behind blind turns and within narrow passages. Baselines that do not combine map information and task relevance can achieve high precision and low FPR mainly because they leave substantial parts of the occluded obstacles unobserved, avoiding false occupied voxels in unseen regions at the cost of low recall. Several baseline loops end in collisions near these unobserved regions. This is especially noticeable in Scene 2, where the baseline maps leave large false-negative gaps inside the L-shaped wall, while the proposed method recovers the occupied voxels behind the corner and is the only method that completes the scene. In Scene 3, the proposed map also captures more of the cluttered occupied voxels that support safe turns.

Fixed and Random do not reliably inspect hidden obstacles

before reaching a turn. Lookahead improves coverage when the path direction exposes them, but does not explicitly target uncertain map regions. NBV-Map often selects outward frontier regions, whereas the proposed method directs observations toward occupied structures relevant to the next maneuver. This increased coverage comes with lower precision and higher FPR in the hard scenes, where localization drift and sensor noise introduce spurious occupied voxels. The results therefore show a tradeoff between recovering previously hidden obstacles and reconstruction precision.

3) *Task-Dependent Map Entropy*: We next evaluate whether the selected views reduce uncertainty in the map regions where the robot needs confidence for safe navigation. This metric is aligned with the proposed formulation: map awareness determines where uncertainty remains, and task weighting determines its relevance to the upcoming maneuver. Fig. 8 reports task-dependent map entropy along the executed base trajectories in every loop. The left panels summarize the distributions over three loops per scene, and the right panels show representative paths colored by local entropy and the final robot configuration when a loop ends in collision.

In Scene 1, all methods complete the task and reduce entropy over repeated loops, while the proposed method achieves lower mean and variance. It also yields lower mean and variance in Scene 2 and Scene 3, where several baselines collide before completing a loop. Failed executions terminate early and reset the map, so they do not accumulate the same uncertainty reduction across traversals. The stronger separation in these scenes is consistent with the reconstruction results. Lookahead helps when the path direction exposes the relevant region, but can leave an occluded turning region unobserved. NBV-Map can instead favor uncertain outward frontiers even when they have little influence on the next maneuver. Task weighting directs the proposed policy toward uncertainty near the upcoming turn, where an additional observation can affect the planned motion.

The right panels of Fig. 8 show the same relation spatially. In Scene 1, all methods complete the task, but the Fixed baseline still leaves the observed map incomplete around the inner turn, visible as a mismatch between the observed map (dark green) and reference map (gray). The corresponding trajectory passes through higher-entropy cells near the blind corner, showing that the entropy metric reflects local map

quality even when the loop succeeds. In Scene 2 and Scene 3, this effect becomes failure-critical. Random leaves the critical turning region unobserved, so the robot plans through the incomplete map toward the next waypoint and collides. NBV-Map shows the complementary failure mode: maintained-map uncertainty is available, but without task weighting the critical turn remains insufficiently observed. In both cases, the final robot configuration lies near the unobserved obstacle region, and the trajectory color shows higher map entropy in that same area. The proposed method observes these regions earlier, so the observed map better matches the reference map and entropy is reduced before the base reaches the turn. Thus, entropy along the path is a representative measure of the local map state that affects navigation safety.

4) *Navigation Safety*: Table II reports the navigation safety benchmark outcomes. The proposed method is the only one that completes all tested loops, whereas each baseline fails in at least one harder scene. Fig. 2 provides representative examples: the red curve is the planned base path, the blue shaded region is the hand camera FoV, and the yellow marker indicates the hidden hazard. In the blind-corner example, the proposed method turns the hand camera toward the corner before the base reaches the turn, so the map can be updated before the robot commits to the maneuver. The random baseline leaves the same region outside the FoV until the base is already close, so the planner continues to rely on an incomplete local map. The office-scene example shows the same behavior around a hidden pole near the turn. These examples show how map- and task-aware viewpoint selection directs the hand camera toward uncertain regions relevant to the next maneuver to enable timely map updates and support safer navigation.

The accompanying video also shows how the selected views change after these hazards have been inspected. Along the straight segment beyond the L-shaped turn, the camera looks outward toward remaining uncertain regions. In the office, it scans the room after checking the hidden pole. Task-aware sensing therefore adapts throughout the maneuver rather than fixing the camera toward one region.

Fig. 9 shows the same effect quantitatively. The dashed red line marks zero clearance based on the controller’s collision model. In Scene 1, the methods are more comparable, although several baselines still approach or cross the zero-clearance boundary; the proposed method maintains one of the highest and most consistent clearance distributions. The separation is clearer in Scene 2, where early observation of the blind turn matters most: the proposed method shifts the distribution

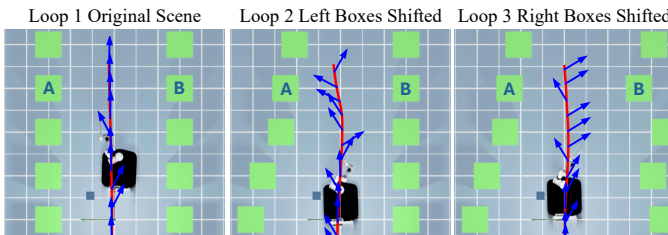
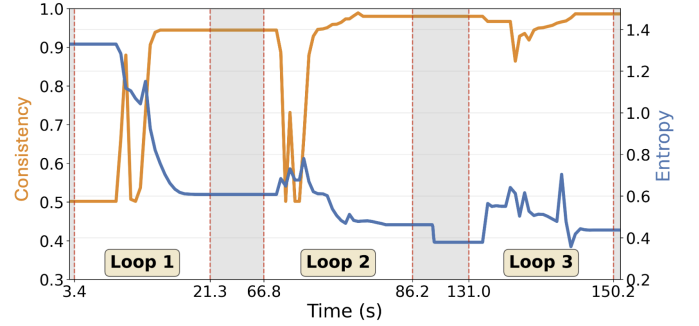
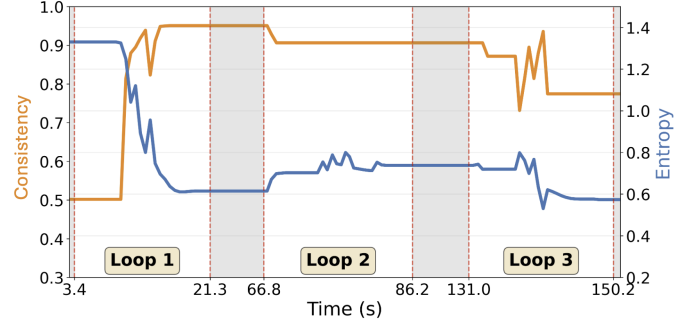


Fig. 10: Controlled Isaac Sim scene-change experiment. The red curves are the executed base trajectories, and the blue arrows show the sampled hand camera headings. After one row of boxes is shifted, the hand camera redirects toward that side to acquire additional observations.



(a) Box A.



(b) Box B.

Fig. 11: Consistency belief and task-dependent map entropy for two representative box regions in the controlled scene-change simulation shown in Fig. 10. Vertical dashed lines mark loop boundaries, and shaded regions indicate intervals when the robot is away from the boxes and receives no observations of them. Each box changes primarily when its side of the corridor is displaced.

upward, while the baselines remain closer to near-contact. In Scene 3, Lookahead completes the loops, but its clearance distribution still contains many near-collision events, indicating little safety margin. The proposed method remains more consistent because it selects views using both the maintained map and the upcoming maneuver. These clearance distributions agree with the qualitative examples and uncertainty analysis.

C. Controlled Simulation Under Scene Changes

The static benchmark demonstrates the benefit of observing missing geometry before a maneuver. We next use a controlled Isaac Sim experiment [60] to examine how the object-aware uncertainty model responds when previously mapped objects move under a fixed navigation task. As shown in Fig. 10, the robot traverses a corridor formed by two rows of $1\text{m} \times 1\text{m}$ boxes over three consecutive loops. Loop 1 starts from an empty map in the reference scene; Loops 2 and 3 start from the previous maintained map. Before Loop 2, boxes on the left are shifted, and before Loop 3, boxes on the right are shifted. The waypoint sequence and simulated environment otherwise remain unchanged, so the experiment focuses on object-level scene changes without the additional variability of the real-world trials. For this simulation, object and free-space confidence decay is turned off so that the observed changes come only from the object displacements.

Fig. 10 shows the qualitative response of the policy. The red curves denote executed base trajectories, and the blue arrows denote sampled hand camera headings. In Loop 1,

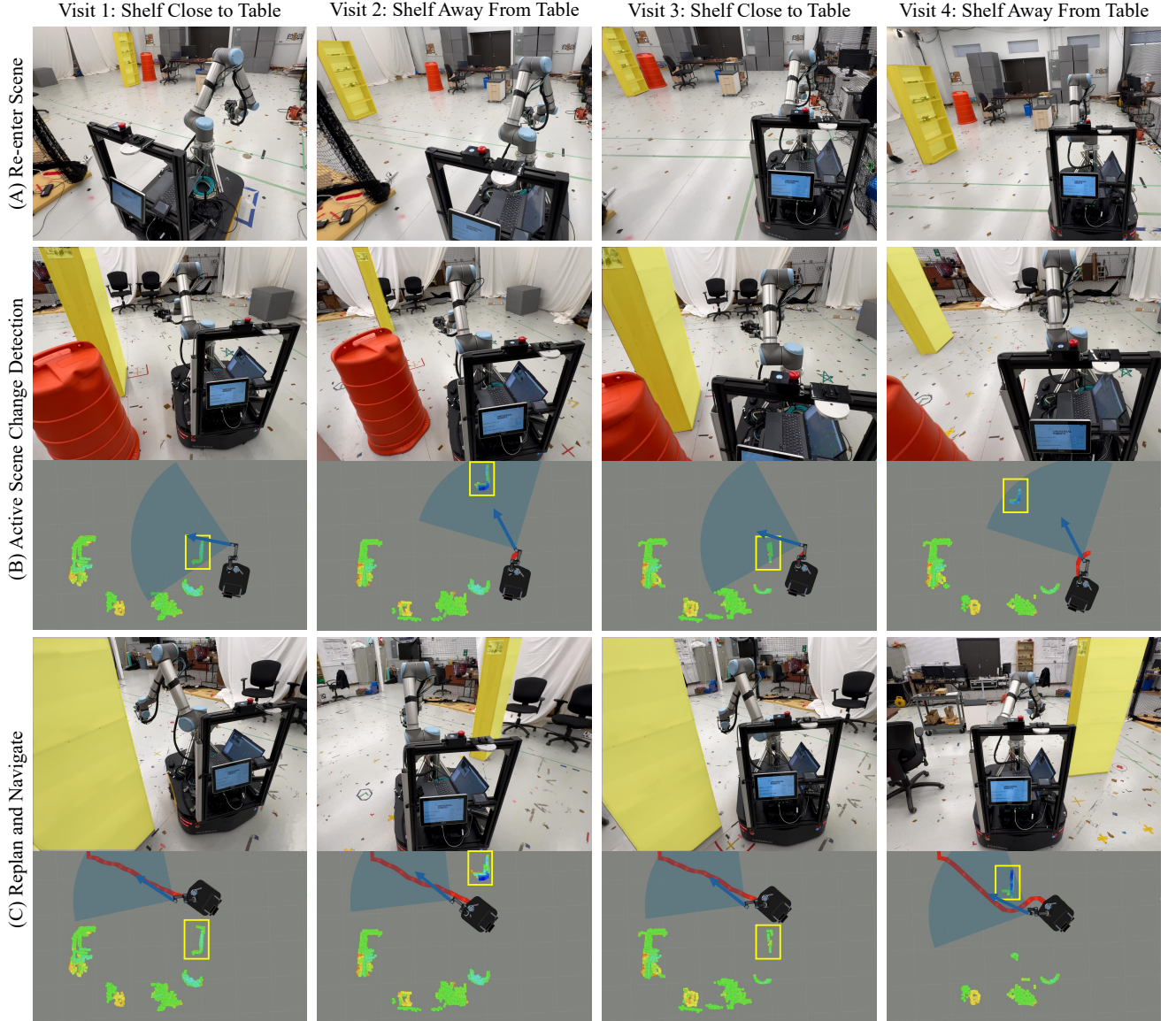


Fig. 12: Real-world long-term scene-change experiment over four visits. In each visit, the robot re-enters the room with the maintained object map, acquires new observations near the shelf, updates the map, and replans according to the current shelf position, switching between the outside and inner routes.

the hand camera largely follows the corridor direction. After the left-side boxes are shifted, the headings in Loop 2 bias toward the left. After the right-side boxes are shifted, the same behavior appears toward the right in Loop 3. Some views remain on the unchanged side because the robot continues the same navigation task, and localization errors and sensor noise can slightly reduce confidence in unchanged objects. Since the base trajectory remains similar across loops, the shift in viewing direction is not explained by the desired path alone. It indicates that inconsistency with the maintained map attracts additional sensing to the changed side, as intended by the object-aware NBV policy.

Fig. 11 quantifies this behavior for two representative boxes. The dashed vertical lines mark loop boundaries, and the shaded regions mark intervals when the robot is away from the boxes and receives no direct observations. The orange curve shows the consistency belief $\mathbb{E}[v_i]$, and the blue curve shows average map entropy over a $1.5\text{m} \times 2.5\text{m}$ horizontal rectangle around the box region, covering both the box and nearby free space

before and after displacement. In Loop 1, the robot starts from an empty map, so the boxes are initialized only after first observation, with consistency belief 0.5 and high associated map entropy. As observations accumulate, both boxes become more consistent with the maintained map. In Loop 2, Box A is displaced: its consistency belief drops and entropy rises when observations first become inconsistent, then the object is re-localized and confidence recovers after more observations. Box B remains comparatively stable. In Loop 3, the dominant change appears in Box B after the right-side displacement; its consistency belief drops, while Box A remains stable. The drop for Box B does not trigger object reconstruction, whereas Box A is re-localized after the earlier displacement. The average entropy around Box B later decreases because moving the box farther from the robot exposes free space in front of it, lowering the regional mean even though the box consistency belief drops. The two objects respond in different loops, showing how map entropy evolves locally as changed objects are observed, consistent with Section V-E.

D. Real-World Long-Term Scene Changes

In the final experiment, we demonstrate the full closed-loop system under real-world long-term scene changes. Fig. 12 shows a four-visit sequence in the office scene (Scene 3 in Fig. 6), where a large shelf is moved closer to or farther from the table between traversals. This qualitative sequence examines how map updates guide sensing and replanning across visits, complementing the static-scene benchmark.

In all four visits, the robot is given the same waypoint sequence. Visit 1 starts from an empty map, and each later visit starts from the accumulated map of the previous visit. Fig. 12 pairs camera images with RViz views of the robot, maintained map, hand camera FoV, and planned base path; the shifted shelf is shaded in yellow. When the shelf is relocated, the maintained map becomes unreliable where the path may need to change. Guided by the proposed object- and task-aware NBV formulation, the robot first inspects free space upon re-entry and then returns to previously mapped objects near the shelf, as seen in the accompanying video. After new observations reveal inconsistency around the relocated shelf, map entropy rises there and attracts additional sensing to refine the map. The robot then replans according to the updated shelf position. In the figure, the hand camera turns toward the shelf region in all visits, while the planned path switches with the shelf position: when the shelf is close to the table in Visits 1 and 3, the robot drives around the shelf from the outside; when the shelf is farther away in Visits 2 and 4, it nudges through from the inner side. Together, the visits demonstrate the active perception and action loop targeted in this work. The robot carries a semi-static object-aware map across traversals, uses task-dependent map entropy to decide where the map should be checked or corrected, and uses the updated map to adapt navigation in a changing environment.

VIII. CONCLUSIONS

We presented an object- and task-aware active perception framework for mobile manipulators operating in cluttered, semi-static environments. The main contribution of the paper is a probabilistic next-best-view formulation that combines maintained map uncertainty, object-level change awareness, and path-dependent task relevance to prioritize observations in regions most likely to affect safe navigation. This formulation is integrated with a hierarchical-task control framework so that the robot can realize informative hand camera motions while preserving the priority of the navigation task.

Real-world experiments demonstrate that the proposed framework reduces task-relevant map uncertainty, improves map consistency, and lowers navigation risk compared with passive or non-task-aware baselines. Overall, the results support the main premise of the paper: in semi-static environments, active perception should not be driven only by visibility or generic information gain, but by which uncertain and potentially changing regions matter most to safe progress along the planned path. By coupling semi-static map maintenance, task-aware viewpoint selection, and whole-body control in closed loop, the proposed system enables safer and more reliable navigation in changing environments across repeated visits.

REFERENCES

- [1] S. Thakar, P. Rajendran, A. M. Kabir, and S. K. Gupta, "Manipulator motion planning for part pickup and transport operations from a moving base," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 1, pp. 191–206, Jan. 2022.
- [2] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.
- [3] L. Schmid, J. Delmerico, J. L. Schonberger, J. Nieto, M. Pollefeys, R. Siegwart, and C. Cadena, "Panoptic multi-TSDFs: A flexible representation for online multi-resolution volumetric mapping and long-term dynamic scene consistency," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 8018–8024.
- [4] J. Qian, V. Chatrath, J. Yang, J. Servos, A. P. Schoellig, and S. L. Waslander, "POCD: Probabilistic object-level change detection and volumetric mapping in semi-static scenes," in *Proc. Robotics: Science and Systems (RSS)*, 2022.
- [5] J. Tordesillas and J. P. How, "PANTHER: Perception-aware trajectory planner in dynamic environments," *IEEE Access*, vol. 10, pp. 22 662–22 677, 2022.
- [6] Y. Zheng, C. Yu, J. Xu, F. Ran, and E. Xu, "Active perception-planning collaboration for quadrotor dynamic obstacle avoidance with NMPC," in *Proc. IEEE International Conference on Unmanned Systems (ICUS)*, 2025, pp. 323–328.
- [7] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proc. IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, 1997, pp. 146–151.
- [8] B. Bogenberger, O. Harrison, O. Dahanagamaarachchi, L. Brunke, J. Qian, S. Zhou, and A. P. Schoellig, "Where did i leave my glasses? open-vocabulary semantic exploration in real-world semi-static environments," *IEEE Robotics and Automation Letters*, vol. 11, no. 3, pp. 3342–3349, Mar. 2026.
- [9] M. Naazare, F. G. Rosas, and D. Schulz, "Online next-best-view planner for 3D-exploration and inspection with a mobile manipulator robot," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3779–3786, Apr. 2022.
- [10] J. Yang, W. Xue, S. Ghavidel, and S. L. Waslander, "Active 6D pose estimation for textureless objects using multi-view RGB frames," *The International Journal of Robotics Research*, Jan. 2026, onlineFirst, doi: 10.1177/02783649251411922.
- [11] H. Ma, M. Shi, B. Gao, and D. Huang, "Active perception for grasp detection via neural graspness field," in *Advances in Neural Information Processing Systems*, vol. 37, 2024, pp. 38 122–38 141.
- [12] L. Di Giammarino, B. Sun, G. Grisetti, M. Pollefeys, H. Blum, and D. Barath, "Learning where to look: Self-supervised viewpoint selection for active localization using geometrical information," in *Computer Vision – ECCV 2024*, ser. Lecture Notes in Computer Science, vol. 15144. Springer, 2025, pp. 188–205.
- [13] M. Hanlon, B. Sun, M. Pollefeys, and H. Blum, "Active visual localization for multi-agent collaboration: A data-driven approach," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 18 034–18 040.
- [14] J. Lim, N. Lawrence, F. Achermann, T. Stastny, R. Bähnmann, and R. Siegwart, "Fisher information based active planning for aerial photogrammetry," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 1249–1255.
- [15] X. Chen, Q. Li, T. Wang, T. Xue, and J. Pang, "GenNBV: Generalizable next-best-view policy for active 3D reconstruction," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 16 436–16 445.
- [16] X. Du, S. Zhou, and A. P. Schoellig, "Hierarchical task model predictive control for sequential mobile manipulation tasks," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1270–1277, Feb. 2024.
- [17] X. Du, J. Qian, S. Zhou, and A. P. Schoellig, "Perceptive hierarchical-task MPC for sequential mobile manipulation in unstructured semi-static environments," *arXiv preprint arXiv:2603.10227*, 2026.
- [18] C. Campos, R. Elvira, J. J. Gómez Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.
- [19] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, Apr. 2013.

- [20] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3D euclidean signed distance fields for on-board MAV planning," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1366–1373.
- [21] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, "SLAM++: Simultaneous localisation and mapping at the level of objects," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 1352–1359.
- [22] M. Rünz, M. Buffier, and L. Agapito, "MaskFusion: Real-time recognition, tracking and reconstruction of multiple moving objects," in *Proc. IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2018, pp. 10–20.
- [23] M. Grinvald, F. Furrer, T. Novkovic, J. J. Chung, C. Cadena, R. Siegwart, and J. Nieto, "Volumetric instance-aware semantic mapping and 3D object discovery," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 3037–3044, Jul. 2019.
- [24] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: an open-source library for real-time metric-semantic localization and mapping," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 1689–1696.
- [25] Q. Gu, A. Kuwajerwala, S. Morin, K. M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa, C. Gan, C. M. de Melo, J. B. Tenenbaum, A. Torralba, F. Shkurti, and L. Paull, "ConceptGraphs: Open-vocabulary 3D scene graphs for perception and planning," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 5021–5028.
- [26] N. Hughes, Y. Chang, and L. Carlone, "Hydra: A real-time spatial perception system for 3D scene graph construction and optimization," in *Proc. Robotics: Science and Systems (RSS)*, 2022.
- [27] J. Cheng, Z. Wang, H. Zhou, L. Li, and J. Yao, "DM-SLAM: A feature-based SLAM system for rigid dynamic scenes," *ISPRS International Journal of Geo-Information*, vol. 9, no. 4, p. 202, Mar. 2020.
- [28] L. Schmid, O. Andersson, A. Sulser, P. Pfrendschuh, and R. Siegwart, "Dynablox: Real-time detection of diverse dynamic objects in complex environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6259–6266, Oct. 2023.
- [29] G. D. Tipaldi, D. Meyer-Delius, and W. Burgard, "Lifelong localization in changing environments," *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1662–1678, Dec. 2013.
- [30] M. Grinvald, F. Tombari, R. Siegwart, and J. Nieto, "TSDF++: A multi-object formulation for dynamic object tracking and reconstruction," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 14 192–14 198.
- [31] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger, "MID-Fusion: Octree-based object-level multi-instance dynamic SLAM," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 5231–5237.
- [32] Y. Ren, B. Xu, C. L. Choi, and S. Leutenegger, "Visual-inertial multi-instance dynamic SLAM with object-level relocalisation," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 11 055–11 062.
- [33] J. Fu, Y. Du, K. Singh, J. B. Tenenbaum, and J. J. Leonard, "NeuSE: Neural SE(3)-equivariant embedding for consistent spatial understanding with objects," in *Proc. Robotics: Science and Systems (RSS)*, 2023.
- [34] S. Looper, J. Rodriguez-Puigvert, R. Siegwart, C. Cadena, and L. Schmid, "3D VSG: Long-term semantic scene change prediction through 3D variable scene graphs," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 8179–8186.
- [35] D. M. Rosen, J. Mason, and J. J. Leonard, "Towards lifelong feature-based mapping in semi-static environments," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1063–1070.
- [36] J. Qian, V. Chatrath, J. Servos, A. Mavrinas, W. Burgard, S. L. Waslander, and A. Schoellig, "POV-SLAM: Probabilistic object-aware variational SLAM in semi-static environments," in *Proc. Robotics: Science and Systems (RSS)*, 2023.
- [37] S. Chen, Y. Li, and N. M. Kwok, "Active vision in robotic systems: A survey of recent developments," *The International Journal of Robotics Research*, vol. 30, no. 11, pp. 1343–1377, Sep. 2011.
- [38] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon "next-best-view" planner for 3D exploration," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1462–1468.
- [39] S. Isler, R. Sabzevari, J. Delmerico, and D. Scaramuzza, "An information gain formulation for active volumetric 3D reconstruction," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 3477–3484.
- [40] Z. Zhang and D. Scaramuzza, "Perception-aware receding horizon navigation for MAVs," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 2534–2541.
- [41] —, "Beyond point clouds: Fisher information field for active visual localization," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 5986–5992.
- [42] —, "Fisher information field: an efficient and differentiable map for perception-aware planning," *arXiv preprint arXiv:2008.03324*, 2020.
- [43] T. Kim and D. Panagou, "Visibility-aware RRT* for safety-critical navigation of perception-limited robots in unknown environments," *IEEE Robotics and Automation Letters*, vol. 10, no. 5, pp. 4508–4515, May 2025.
- [44] A. Mollaei Khass, G. Liu, V. Pandey, W. Jiang, B. Lei, K. Daniilidis, and N. Motee, "Active next-best-view optimization for risk-averse path planning," *arXiv preprint arXiv:2510.06481*, 2025.
- [45] B. Burgess-Limerick, C. Lehnert, J. Leitner, and P. Corke, "An architecture for reactive mobile manipulation on-the-move," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 1623–1629.
- [46] J. Pankert and M. Hutter, "Perceptive model predictive control for continuous mobile manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6177–6184, Oct. 2020.
- [47] M. Spahn, B. Brito, and J. Alonso-Mora, "Coupled mobile manipulation via trajectory optimization with free space decomposition," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 12 759–12 765.
- [48] S. Xie, C. Hu, D. Wang, J. Johnson, M. Bagavathiannan, and D. Song, "Coupled active perception and manipulation planning for a mobile manipulator in precision agriculture applications," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 12 665–12 671.
- [49] S. Jauhari, S. Lueth, and G. Chalkatzaki, "Active-perceptive motion generation for mobile manipulation," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 1413–1419.
- [50] S. Uppal, A. Agarwal, H. Xiong, K. Shaw, and D. Pathak, "SPIN: Simultaneous perception, interaction and navigation," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 18 133–18 142.
- [51] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, Feb. 1987.
- [52] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006–1028, Jun. 2014.
- [53] J. Qian, S. Zhou, N. J. Ren, V. Chatrath, and A. P. Schoellig, "Closing the perception-action loop for semantically safe navigation in semi-static environments," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 11 641–11 648.
- [54] G. Vogiatzis and C. Hernández, "Video-based, real-time multi-view stereo," *Image and Vision Computing*, vol. 29, no. 7, pp. 434–441, Jun. 2011.
- [55] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs, NJ, USA: Prentice Hall, 1993, vol. 1.
- [56] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Hoboken, NJ, USA: Wiley-Interscience, 2006.
- [57] Z. Liao, J. Yang, J. Qian, A. P. Schoellig, and S. L. Waslander, "Uncertainty-aware 3D object-level mapping with deep shape priors," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 4082–4089.
- [58] B. Siciliano, L. Sciacivco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, 1st ed. London, U.K.: Springer, 2009, ch. 11.2.
- [59] R. Verschuere, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "acados—a modular open-source framework for fast embedded optimal control," *Mathematical Programming Computation*, vol. 14, no. 1, pp. 147–183, Mar. 2022.
- [60] NVIDIA, "Isaac Sim: Robotics simulation and synthetic data generation," <https://developer.nvidia.com/isaac/sim>, accessed: Oct. 1, 2026.