VLM-GROUNDED TASK AND MOTION PLANNING WITH UNCERTAINTY AWARE ACTIVE PERCEPTION

BY

AUSTINE OLOO

BS, Gretsa University, 2023

THESIS

Submitted in partial fulfillment of the requirements for
the degree of Masters of Science in Computer Science
in the Graduate School of
Binghamton University
State University of New York
2025

Accepted in partial fulfillment of the requirements for
the degree of Masters of Science in Computer Science
in the Graduate School of
Binghamton University
State University of New York
2025

December 10, 2025

Shiqi Zhang, Chair and Faculty Advisor
School of Computing, Binghamton University

Monika Roznere, Member
School of Computing, Binghamton University

Abstract

Task and Motion Planning (TAMP) provides a principled framework for long-horizon robotic manipulation, yet classical approaches assume fully observable, static environments, assumptions that fail catastrophically in real-world deployment. Vision-Language Models (VLMs) offer a compelling solution by grounding symbolic predicates in open-vocabulary visual observations, enabling TAMP systems to reason about previously unknown objects and scenes. However, current VLM-TAMP systems treat these models as infallible oracles: queried once, trusted implicitly, and never questioned. VLMs provide no reliable signal when their perceptions are uncertain and offer no mechanism to actively resolve ambiguity; when faced with occlusion, clutter, or dynamic scene changes, these systems either fail silently or propagate perceptual errors through subsequent execution. This thesis introduces **VLM-grounded Task and Motion Planning with uncertainty-aware active perception (VAP-TAMP)**, a closed-loop framework that transforms VLMs from passive perception modules into active participants in robust task execution while handling unforeseen situations. We integrate PDDL-based symbolic planning with VLM-powered semantic grounding through three contributions: (1) a Multi-Query Consistency based Uncertainty Detector (MQCD) that identifies unreliable VLM responses by probing the model with semantically equivalent queries and measuring response agreement; (2) an improved entropy-based semantic viewpoint selection mechanism, for active perception that maximizes expected reduction in VLM belief entropy over queried predicates, approximated through a geometric proxy that prioritizes viewpoints revealing previously occluded object features; and (3) a situation-aware replanning

mechanism, that leverages VLM-based post-action verification to detect execution failures and environmental changes, initiating PDDL-based recovery planning. We evaluate on a real-world mobile manipulator across tasks involving occlusion, adversarial disturbance, and safety-critical state verification, significantly outperforming state-of-the-art baselines and demonstrating that explicitly quantifying and actively resolving VLM uncertainty is essential for deploying VLM-TAMP systems in unstructured, human-centric environments.

This thesis is dedicated to the loving memory of my grandpa, Ngaji Akungu, and to my uncle, Nixon Ngaji, whose guidance, strength, and enduring support, along with that of my entire family, have shaped this journey.

Acknowledgements

I would like to express my deepest gratitude to my advisor, Professor Shiqi Zhang, for his invaluable guidance and mentorship throughout this research. His expertise in artificial intelligence and robotics inspired me to explore this field and helped shape the foundation and direction of this work. I am truly grateful for his patience, encouragement, and continuous support at every stage of my academic journey.

I also extend my sincere appreciation to Professor Monika Roznere for serving as a committee member during my thesis defense.

I would also like to acknowledge that some materials utilized in this thesis were adapted from the DK prompt work by Binghamton University AIRLAB, which provided a valuable foundation for this thesis.

I also extend my sincere appreciation to my collaborator, Xinwei Guo, who supported the project in many aspects.

AI Usage Statement

I certify that I have used generative AI tools or services for any aspect of my thesis. I used Claude AI to to polish the grammar and improve the clarity of the text throughout the manuscript.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Task and Motion Planning (TAMP) has long provided a principled framework for long horizon robotic manipulation, combining symbolic reasoning with geometric motion planning to solve complex, multi step tasks [1]. Classical TAMP systems offer formal guarantees and compositional generalization, yet they operate under restrictive assumptions: fully observable environments, static scenes, and known object models. These assumptions fail catastrophically in real-world deployment, where robots must contend with occlusion, clutter, dynamic changes, and previously unseen objects.

Vision-Language Models (VLMs) have emerged as a compelling solution to this limitation. By grounding symbolic predicates in open-vocabulary visual observations, VLMs enable TAMP systems to perceive and reason about previously unknown objects and scenes [2–4]. A robot can now interpret commands like "bring me something to drink coffee with" and visually verify whether "the mug is on the table." This capability has sparked a new generation of VLM-TAMP systems that promise to operate in previously inaccessible unstructured environments. Yet a fundamental gap remains. Current VLM-TAMP systems treat these powerful models as infallible oracles: queried once, trusted implicitly, and never questioned [5–8]. This assumption is problematic. VLMs frequently fail in precisely the conditions that define real-world deployment: occluded objects, cluttered scenes, ambiguous viewpoints, and

dynamic environments [9, 10]. When a VLM misidentifies a bowl as a mug due to partial occlusion, or fails to notice that an object has been moved, the error propagates silently through subsequent planning and execution. Current systems provide no reliable signal when perception is uncertain, no mechanism to actively resolve ambiguity, and no capacity to recover when actions fail. The result is brittle autonomy, an impressive demonstrations that collapse under the adversarial conditions of real world operation.

We argue that the path to robust mobile manipulation requires a paradigm shift: from passive, single-shot perception to active, uncertainty-aware perception-action loops. The robot must know when it does not know, actively seek disambiguating information, and verify that its actions achieve their intended effects. The key insight underlying this work is that VLM uncertainty, rather than being a nuisance to suppress, is a valuable signal that should trigger active information gathering. When a VLM is uncertain whether an object is a mug or a bowl, the robot should not guess, it should move to obtain a better viewpoint. When post-action verification reveals that a grasp failed, the robot should not proceed blindly, it should replan.

This thesis introduces **VLM-grounded Task and Motion Planning with uncertainty-aware active perception (VAP-TAMP)**, a closed-loop framework that transforms VLMs from passive perception modules into active participants in robust task execution while handling unforeseen situations. VAP-TAMP integrates classical PDDL-based symbolic planning with VLM-powered semantic grounding, enabling robots to detect perceptual uncertainty, actively resolve ambiguity through viewpoint exploration, and recover from execution failures through symbolic replanning.

This thesis makes three contributions:

1. A **Multi-Query Consistency based Uncertainty Detector (MQCD)** that identifies unreliable VLM responses by probing the model with semantically equivalent queries and measuring response agreement. Rather than trusting a single query, MQCD gener-

ates multiple syntactically distinct but semantically equivalent prompts and computes a consistency score. Low consistency reliably indicates genuine perceptual ambiguity, identifying scenes where single-shot perception is likely to fail.

2. An **improved entropy-based semantic viewpoint selection mechanism** for active perception that maximizes expected reduction in VLM belief entropy over queried predicates, approximated through a geometric proxy that prioritizes viewpoints revealing previously occluded object features. This approach is grounded in the insight that semantic ambiguity typically arises from geometric occlusion.

3. A **situation-aware replanning mechanism** that leverages VLM-based post-action verification to detect execution failures and environmental changes, initiating PDDL-based recovery planning. After each manipulation primitive, the VLM verifies whether expected effects were achieved. Detected mismatches, such as a failed grasp, a displaced object, or an unexpectedly closed door, trigger symbolic state updates and recovery planning, enabling the robot to adapt rather than fail.

The remainder of this thesis is organized as follows. Chapter 2 reviews related work in Task and Motion Planning, Vision-Language Models for robotics, and active perception, positioning VAP-TAMP within the current research landscape. Chapter 3 presents the VAP-TAMP methodology in detail, including MQCD for uncertainty quantification, entropy-based semantic viewpoint selection, and the situation-aware replanning algorithm. Chapter 4 describes our experimental setup, baseline comparisons, and comprehensive results. Chapter 5 concludes with a discussion of limitations and directions for future research.

Chapter 2

Related Work

This work sits at the intersection of classical Task and Motion Planning (TAMP), modern Vision-Language Models (VLMs) for robotic perception, and active perception strategies for handling uncertainty. While significant progress has been made in each of these areas independently, the integration of VLM-based perception with classical TAMP frameworks, particularly with active perception mechanisms to address VLM uncertainty still remains largely unexplored. This chapter reviews the relevant literature across these domains, weaving together insights that motivate and contextualize our approach.

2.1 Task and Motion Planning

Task and Motion Planning (TAMP) addresses the fundamental challenge of combining discrete, symbolic reasoning about tasks with continuous motion planning in physical space [1, 11]. The goal is to synthesize plans that are both symbolically valid for achieving high level goals and kinematically feasible for robot execution. Early TAMP systems employed hierarchical decomposition, separating symbolic planning from geometric reasoning [12]. These approaches typically use symbolic planners such as PDDL-based systems like FastForward [13] to generate

high-level action sequences, followed by motion planners including RRT [14] and PRM [15] to compute collision-free trajectories. However, this sequential approach suffers from a fundamental limitation: symbolic plans may prove motion-infeasible, requiring expensive backtracking and replanning cycles that substantially degrade system performance.

To address this inefficiency, more tightly integrated approaches have emerged over the past decade. Kaelbling and Lozano-Pérez [11] introduced frameworks that interleave symbolic and geometric reasoning, where motion planning failures directly inform the symbolic search space. Building on this foundation, PDDLStream [16] extends the idea by unifying streams of geometric samples with symbolic planning, enabling more efficient exploration of the combined discrete-continuous search space. Similarly, SMTPlan [17] and its variants leverage constraint satisfaction techniques to tightly couple symbolic and geometric constraints, demonstrating impressive performance in benchmark domains.

Recent work has begun addressing the visual grounding challenge in TAMP. GROP [18] introduced visually grounded task and motion planning, enabling robots to ground symbolic predicates in visual observations for manipulation tasks. This approach was subsequently extended to incorporate large language models in LLM-GROP [19], which leverages LLMs to bridge natural language instructions with visually grounded planning, enabling more flexible human-robot interaction. These works provide foundational insights that inform our approach to integrating VLM-based perception with symbolic planning.

Despite these advances, classical TAMP frameworks share a critical assumption that becomes their primary limitation in real-world deployment: the world state is fully known and static. Symbolic predicates such as `on(block, table)` or `clear(cup)` are assumed to be accurately maintained throughout execution, with perfect correspondence to physical reality. This assumption holds reasonably well in controlled laboratory environments equipped with engineered perception systems (fiducial markers, depth cameras) paired with known object

models, or structured lighting setups. However, it breaks down dramatically when robots operate in real-world human environments, which are characterized by partial observability (objects occluded or outside sensor field of view), dynamic changes during task execution due to human activities or physical interactions, and inherent perceptual uncertainty in object recognition and pose estimation, especially for novel or texture-less objects.

Some works have attempted to incorporate uncertainty into TAMP through probabilistic extensions. T-RRT [20] and its variants handle continuous state uncertainty through probabilistic motion planning, while belief-space TAMP approaches [21, 22] maintain distributions over possible world states and reason about expected outcomes. However, these methods still fundamentally rely on accurate *a priori* object models and known uncertainty distributions. More critically, they struggle to scale to open-vocabulary scenarios where object classes are not pre-specified and must be discovered at runtime.

Our approach fundamentally advances beyond these limitations by replacing the assumption of perfect state knowledge with VLM-based perception capable of open-vocabulary understanding, while introducing active perception to detect and resolve uncertainty before planning. Unlike belief-space TAMP methods that attempt to plan optimally under all possible belief states which scales poorly in high dimensional, open-vocabulary domains, we proactively improve state estimates through targeted information gathering, ensuring that classical TAMP proceeds with sufficiently confident observations. This design philosophy trades the theoretical completeness of probabilistic planning for practical scalability and computational efficiency in real-world, unstructured environments where the space of possible objects and configurations cannot be enumerated a priori.

2.2 Vision-Language Models for Robotic Perception and Planning

The emergence of large-scale Vision-Language Models (VLMs) such as Gemini [23], CLIP [24], Flamingo [25], and GPT-4V [26] has fundamentally transformed the landscape of robotic perception. Pretrained on massive internet scale image-text datasets, these models exhibit remarkable zero-shot capabilities in visual understanding, object recognition, and spatial reasoning, offering a compelling alternative to traditional perception pipelines that require extensive task-specific training data and predefined object categories.

Recent work has demonstrated that VLMs can serve as powerful scene understanding modules for robotic systems [8, 27, 28]. SayCan [8] leverages VLMs to compute affordance probabilities by determining which actions are feasible given the current scene configuration and uses these probabilities to guide a symbolic planner toward executable action sequences. This effectively prunes the search space, focusing planning resources on actions likely to succeed. In a complementary direction, Code as Policies [7] uses large language models to generate executable Python code for robotic manipulation, grounding abstract natural language commands in concrete robot APIs. Other approaches, including GroundedSAM [29] (combining Grounding DINO [30] with Segment Anything [31]), VILD [32], and GLIPv2 [33], enable open-vocabulary object localization and segmentation from free-form text queries by aligning visual and language embeddings in a shared semantic space.

Beyond perception, VLMs and large language models have been directly integrated into task planning pipelines. Inner Monologue [34] uses an LLM to generate high-level plans and iteratively refines them based on environmental feedback, including success and failure detection signals. LLM-TAMP [35] generates PDDL-style action sequences conditioned on scene descriptions, demonstrating that language models can capture commonsense reasoning

about task structure and temporal dependencies. LLM+P [36] takes a complementary approach by using LLMs to translate natural language task specifications into PDDL problem definitions, leveraging the formal guarantees of classical planners while benefiting from LLMs' natural language understanding capabilities. This separation of concerns (using LLMs for problem specification and classical planners for plan synthesis) offers advantages in terms of plan correctness and interpretability. PaLM-E [5] and RT-2 [37] push this integration further by training end-to-end visuomotor policies that map directly from images and language to robot actions, effectively bypassing explicit symbolic planning in favor of learned reactive behaviors.

The challenge of open-world planning, where robots must handle novel objects and scenarios not seen during training has motivated several recent approaches. COWP [38] addresses open-world task planning by leveraging LLMs to reason about novel objects and generate task plans that generalize beyond predefined object categories. This work demonstrates that LLMs can provide commonsense knowledge about object affordances and task structure even for objects not explicitly represented in the planning domain. DKPrompt [39] advances open-world planning further by using computer vision models to detect and reason about novel objects in the scene, automatically incorporating discovered objects into the planning domain. These approaches share our motivation of enabling robots to operate in open-vocabulary settings, though they do not explicitly address the uncertainty inherent in VLM-based perception.

While these approaches showcase impressive capabilities, a growing body of evidence reveals systematic failure modes in VLM-based systems that have received insufficient attention in the robotics community. VLMs struggle with precise spatial reasoning for tasks requiring understanding of fine-grained spatial relationships such as "the cup is to the left of the book" or accurate counting [40]. They exhibit hallucination behaviors, confidently reporting objects that are not present in the scene [41]. Occlusion handling remains problematic: when objects are partially occluded or outside the field of view, VLMs frequently fail to detect them or

report incorrect attributes [42]. Perhaps most concerning for robotic applications, many VLMs produce deterministic outputs without calibrated confidence scores, providing no signal about when their predictions should be trusted or questioned [43].

Critically, most existing VLM-based robotic systems treat VLM outputs as ground truth, directly incorporating them into planning and control without accounting for uncertainty. When integrated into TAMP frameworks, incorrect VLM predictions propagate through the symbolic reasoning layer, leading to infeasible plans, wasted computational resources, or outright task failures during execution. A few recent works have begun acknowledging this issue: VoxPoser [6] uses value maps to capture spatial uncertainty in affordance predictions, and VLMbench [44] proposes systematic benchmarks for evaluating VLM reliability in robotic contexts. However, these approaches remain fundamentally passive, they measure or represent uncertainty but do not actively resolve it through targeted perception actions.

Our work represents a paradigm shift from passive uncertainty representation to active uncertainty resolution. By explicitly modeling VLM uncertainty and introducing a verification and active perception mechanism, we autonomously trigger information-gathering behaviors when the VLM exhibits uncertainty about critical scene properties. This active approach offers three key advantages over passive methods: first, it prevents uncertain VLM predictions from propagating into planning, eliminating wasted computation on infeasible plans; second, it systematically improves state estimate quality through targeted sensing actions rather than hoping for fortuitous initial observations; third, it ensures that planning proceeds only with adequately confident state estimates, dramatically reducing task failure rates. Where passive approaches accept VLM uncertainty as an unavoidable limitation, we transform it into an actionable signal that drives intelligent sensing behavior.

2.3 Active Perception and Information Gathering

Active perception is the deliberate control of sensors or robot motion to acquire maximally informative observations [45, 46] has long been recognized as essential for robust operation in partially observable environments where initial sensor data is insufficient to uniquely determine the world state. Early active perception systems focused on view planning for object recognition and pose estimation, with Next-Best-View (NBV) algorithms [47, 48] computing camera poses that maximize expected information gain, typically formalized using entropy reduction or mutual information metrics [49]. In simultaneous localization and mapping (SLAM), active perception strategies select robot trajectories that minimize map uncertainty while ensuring exploration coverage [50].

Recent work has extended active perception to manipulation settings through interactive perception [46], where robot actions such as pushing, poking, or grasping serve as information-gathering strategies to reveal hidden objects or disambiguate object properties. Methods including Driven by Learning [51] and related approaches [52] learn policies for singulating objects in cluttered scenes by physically interacting with the environment, using the resulting scene changes to improve state estimates.

Integrating active perception with task planning presents substantial challenges because it requires reasoning jointly about when to gather information versus when to act toward the goal. Contingent planning frameworks [53, 54] represent plans as decision trees with observation nodes, explicitly encoding information-gathering actions and conditional branches based on observation outcomes. However, these approaches suffer from exponential branching as the planning horizon increases, making them difficult to scale beyond toy problems. Belief-space planning methods [21, 55] model uncertainty explicitly using Partially Observable Markov Decision Processes (POMDPs), where the robot maintains a belief distribution over possible world states and information-gathering actions are naturally incentivized when they

reduce uncertainty and enable higher expected task reward. While theoretically elegant, POMDP solvers face severe computational challenges with high-dimensional state spaces and long task horizons, both of which are ubiquitous in manipulation domains.

A more pragmatic approach adopted by many deployed systems is open-loop execution with replanning: execute a plan based on current state estimates, detect failures during execution through monitoring, and replan when failures occur [56]. FF-Replan [57] and related methods employ this strategy, trading optimality for computational tractability. However, they require robust failure detection mechanisms and may waste resources on infeasible plans before triggering replanning.

Existing active perception methods remain largely decoupled from modern VLM-based perception systems, representing a significant missed opportunity for synergy. Classical NBV algorithms assume parametric sensor models, known camera intrinsics, depth uncertainty models derived from sensor physics and optimize over low-level geometric quantities without access to semantic-level reasoning. Interactive perception methods focus on geometric disambiguation, such as revealing occluded surfaces through physical interaction, but do not leverage the rich semantic understanding that VLMs provide to determine what information is most critical to disambiguate for successful task completion. This disconnect between traditional active perception's geometric focus and VLMs' semantic capabilities leaves a gap in handling the types of uncertainty that VLMs introduce: semantic ambiguity about object identity, attribute confusion, or incomplete scene understanding due to limited viewpoints.

This work bridges this gap through a novel integration where VLM-based perception and active perception form a tightly-coupled feedback loop, yielding capabilities that neither approach achieves independently. We use VLM uncertainty scores(derived from response consistency across semantically equivalent queries) as direct triggers for active perception behaviors, creating a system where semantic uncertainty drives geometric sensing decisions. This

integration offers distinct advantages: unlike classical NBV methods that optimize viewpoints based solely on geometric information gain, our approach prioritizes viewpoints that resolve semantically critical uncertainties identified by the VLM (e.g., disambiguating between visually similar objects that have different manipulation affordances). Unlike interactive perception methods that operate without semantic guidance, our approach uses VLM reasoning to determine which physical interactions will yield the most task-relevant information. When uncertainty is detected about scene properties critical for task success, such as object presence due to occlusion or confusion between similar objects, the system autonomously plans and executes information-gathering actions before committing to a manipulation plan. This semantic-geometric coupling ensures that symbolic planning operates on state estimates of sufficient quality while focusing sensing resources on uncertainty that actually matters for task completion, rather than pursuing comprehensive but task-irrelevant information gathering.

2.4 Failure Recovery and Replanning in Robotic Systems

Robust execution in real-world environments demands mechanisms to detect failures and generate alternative courses of action when plans go awry. Failure recovery has been extensively studied across classical planning and robot control communities, yielding diverse strategies that trade off between computational efficiency and solution quality. Replanning approaches generally fall into three categories: plan repair methods [58] that modify existing plans to account for new information while preserving valid plan segments, full replanning strategies [59] that discard current plans and compute new ones from scratch using updated world states, and hybrid approaches [60] that attempt fast plan repair when possible and fall back to full replanning when necessary. In robotics, replanning is typically triggered by sensory feedback indicating unexpected outcomes (a failed grasp, an object not in the

expected location, or collision detection) prompting the system to reestimate the world state and generate a new plan [61].

Existing replanning systems implicitly assume access to ground truth or high-fidelity state estimation, typically provided through accurate object tracking, calibrated sensors, or engineered perception pipelines. This assumption becomes problematic in VLM-based planning systems for several reasons. First, VLMs may provide incorrect scene interpretations from the outset, leading to infeasible plans before execution even begins. Second, execution failures may stem from perceptual errors rather than motion failures, if the VLM incorrectly identifies an object's location or attributes, the resulting grasp or navigation plan will fail regardless of motion planning quality. Third, naively replanning with the same incorrect VLM output will inevitably lead to repeated failures, wasting resources and potentially damaging objects or the environment.

Our approach advances beyond traditional replanning by explicitly distinguishing between perceptual failures and motion failures, enabling targeted recovery strategies that existing methods cannot achieve. When VLM uncertainty is detected during the planning phase, we employ active perception to improve state estimates before generating plans, preventing perceptual errors from propagating into execution. When execution failures occur despite this precaution, we use VLM-based verification to diagnose the failure source: if post-execution VLM queries reveal that the world state differs from predictions, we trigger active perception to correct the state estimate before replanning, addressing the root cause rather than merely generating alternative plans. This failure-aware recovery strategy offers two critical advantages over traditional replanning: first, it prevents the repeated failure cycles that plague VLM-based systems when perceptual errors are not explicitly addressed; second, it efficiently allocates recovery resources by distinguishing between perception-level and motion-level failures. Coupled with our uncertainty quantification and viewpoint selection mechanisms, this provides a comprehensive framework for robust task execution that explicitly handles

the unique challenges introduced by VLM uncertainty in ways that classical replanning approaches fundamentally cannot.

Chapter 3

Methodology

This chapter details the methodology behind **VLM-grounded Task and Motion Planning with uncertainty-aware active perception (VAP-TAMP)**. We present the formal problem definition, the system architecture, and the algorithms governing navigation, manipulation, and closed-loop execution with VLM-based verification.

3.1 System Overview

VAP-TAMP addresses a fundamental challenge in deploying Task and Motion Planning (TAMP) systems to real-world mobile manipulation: the perception-state grounding problem. While symbolic planners operate over abstract predicates assuming perfect state observability, real-world execution requires continuously verifying that these symbolic representations accurately reflect physical reality. This gap between symbolic assumptions and perceptual uncertainty is the core problem VAP-TAMP is designed to solve.

Consider a robot tasked with retrieving a mug from a microwave. A TAMP system generates a plan with actions like `open(microwave)`, `grasp(mug)`, and `place(mug, counter)`. Each action has preconditions: `grasp(mug)` requires that the mug be visible and reachable. Traditional

TAMP assumes these predicates can be evaluated with certainty. In practice, the robot's camera may provide an ambiguous view (e.g., is the microwave door fully open or only partially? Is that cylindrical object the target mug or a similar container?). These perceptual ambiguities can cause plan execution to fail silently or catastrophically.

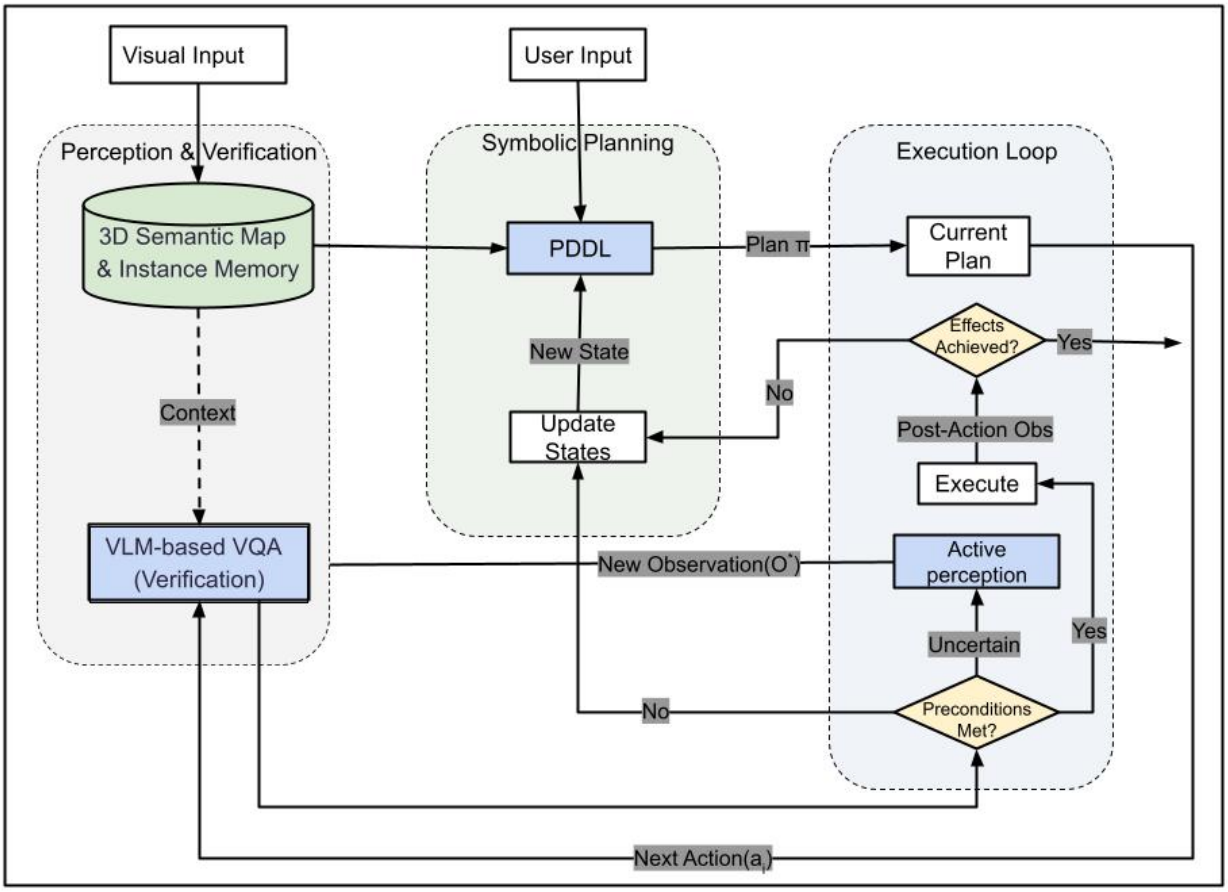


Figure 3.1: The VAP-TAMP system architecture. The framework comprises three core layers: (1) A Symbolic Planning Layer that generates high-level PDDL plans under standard assumptions; (2) A Perception Layer (our core contribution) utilizing VLM-based verification and 3D semantic mapping to ground symbolic predicates in physical observations; and (3) An Execution Layer managing robot control. Two novel feedback loops are highlighted: the Active Perception Loop powered by MQCD and entropy-based viewpoint selection for resolving uncertainty before action execution, and the Situation Handling Loop for error recovery after execution failures.

VAP-TAMP closes this gap by introducing active perception as a first class component of plan execution. Rather than passively accepting initial observations, VAP-TAMP treats perceptual uncertainty as a signal to gather additional information through purposeful robot

motion. The system integrates three layers that work together to ground symbolic plans in physical reality:

- **Symbolic Planning Layer:** Generates high-level action sequences using PDDL planning. This layer operates under the standard TAMP assumption of perfect state knowledge. VAP-TAMP does not modify the planning formalism itself but rather addresses the gap between planning assumptions and execution reality.
- **Perception Layer:** Provides VLM-based verification to ground symbolic predicates in real observations. This layer is responsible for answering the question: "Does the current physical state satisfy the symbolic preconditions required for the next action?" When the answer is uncertain, this layer triggers active perception to resolve ambiguity.
- **Execution Layer:** Handles low-level motion control for navigation and manipulation. This layer executes both planned actions and the information-gathering motions required by active perception.

3.2 Problem Formulation

We formalize the task as a plan execution problem under perceptual uncertainty. This formulation explicitly captures the gap between symbolic planning assumptions and the partial observability inherent in real-world robotic perception.

Planning Domain

The underlying planning problem is defined over a tuple $\Sigma = \langle \mathcal{P}, \mathcal{A}, s_0, g \rangle$, where $\mathcal{P}$ is a set of predicates describing world properties, $\mathcal{A}$ is a set of parameterized actions, s_0 is the initial

symbolic state, and g is the goal condition. Each action $a \in \mathcal{A}$ is defined by preconditions $\text{Pre}(a)$ and effects $\text{Eff}(a)$. In the idealized TAMP setting, a plan $\pi = \{a_1, a_2, \ldots, a_n\}$ is valid if:

$$\forall i \in \{1, \ldots, n\}, \quad \text{Pre}(a_i) \subseteq s_{i-1} \quad \text{and} \quad s_i = (s_{i-1} \setminus \text{Eff}^-(a_i)) \cup \text{Eff}^+(a_i) \tag{3.2.1}$$

This formulation assumes the symbolic state s is known with certainty at each step. VAP-TAMP does not modify this planning formalism; we use standard PDDL planners as black boxes. Our contribution lies in how we verify and maintain alignment between the symbolic state and physical reality during execution.

The Perception-State Grounding Problem

In real-world execution, the robot cannot directly access the true state s_{real}. Instead, it receives observations o from onboard sensors (RGB-D cameras) that provide only partial information about the world. We formalize this through a VLM-based observation function V that maps an image observation o and a symbolic predicate p to a ternary evaluation:

$$V(o, p) \to \{\text{True}, \text{False}, \text{Uncertain}\} \tag{3.2.2}$$

The inclusion of *Uncertain* as an explicit output is crucial. Unlike binary classifiers that force a decision even with low confidence, our formulation acknowledges that some observations are genuinely ambiguous. This uncertainty can arise from:

- **Occlusion:** The target object is partially hidden by other objects or the environment.
- **Viewpoint limitations:** The current camera angle does not reveal discriminative features.

- **Sensor noise:** Poor lighting conditions or motion blur degrade image quality.
- **Semantic ambiguity:** Similar objects that are difficult to distinguish from the current view.

The core guarantee VAP-TAMP provides is: before executing action a_i, the system ensures that for all predicates $p \in \text{Pre}(a_i)$, the observation function returns $V(o, p) = \text{True}$ with high confidence. When this condition cannot be satisfied from the current viewpoint, VAP-TAMP autonomously moves to gather additional observations until uncertainty is resolved.

Theoretical Guarantees

We provide theoretical grounding for the active perception mechanism, establishing conditions under which VAP-TAMP is guaranteed to resolve uncertainty.

Theorem 3.2.1 (Probabilistic Completeness)**.** *Given a finite set of reachable viewpoints $\mathcal{V}$ and a VLM oracle with non-zero probability of correct classification $P(\text{correct}|v) > 0.5$ for at least one viewpoint $v \in \mathcal{V}$, the active perception process terminates with the correct predicate evaluation with probability approaching 1 as the number of viewpoints explored increases.*

Proof Sketch. Let the true state of a predicate be y^*. At each step t, the robot selects a viewpoint v_t and receives a noisy observation y_t from the VLM. We model the VLM response as a Bernoulli random variable with success parameter p_v. Assuming observations are conditionally independent given the state (a mild assumption for viewpoints with sufficient spatial separation), the aggregated decision follows the Law of Large Numbers. As $N \to \infty$, the accumulated evidence (e.g., via majority voting or Bayesian updates) concentrates probability mass on the ground truth y^*. Since the algorithm prioritizes viewpoints by information gain, it effectively samples from the most informative subset of $\mathcal{V}$, accelerating convergence compared to random sampling. □

Complexity Analysis. The computational overhead of active perception is bounded:

- **Viewpoint Generation:** $O(|M| \log |M|)$ for a voxel map of size $|M|$, dominated by ray-casting and sorting operations. This is efficient for typical room-scale maps.
- **VLM Queries:** $O(k)$ where k is the maximum number of viewpoints explored. Since VLM inference takes constant time C_{vlm}, and in practice $k \leqslant 5$, this adds negligible overhead compared to physical motion time.
- **Replanning:** $O(|A|^d)$ where $|A|$ is action space size and d is plan depth. By maintaining valid plan prefixes, VAP-TAMP minimizes d for recovery planning, ensuring replanning is significantly faster than planning from scratch.

3.3 Mapping and Scene Understanding

The robot navigates using a hybrid map comprising a 2D occupancy grid M_{2D} for localization and a 3D semantic voxel map M_{3D} for object grounding.

3D Semantic Mapping and Instance Memory

The environment is represented as a voxel grid $\mathcal{V}$. Each voxel $v(x, y, z)$ stores its occupancy state, color, and a semantic feature embedding. The voxel map is constructed from RGB-D observations using volumetric fusion. Voxels are updated via Truncated Signed Distance Function (TSDF) integration:

$$v_{occ} \leftarrow \text{TSDF\_update}(v, D_t, T_t) \tag{3.3.1}$$

$$v_{color} \leftarrow \text{weighted\_average}(v_{color}, I_t) \tag{3.3.2}$$

Object Instance Segmentation: To identify manipulable objects, we compute a cosine similarity score between a voxel's semantic embedding v_{sem} and the text embedding e_q of a query q:

$$\text{score}(v) = \frac{v_{sem} \cdot e_q}{\|v_{sem}\|\|e_q\|} \tag{3.3.3}$$

High-scoring voxels are clustered using DBSCAN to form object instances I_k. The centroid is computed as centroid(I_k).

Instance Memory: The system maintains a memory $\mathcal{M}$ of tracked objects. New detections are matched to existing entries using the Hungarian algorithm on Euclidean distance:

$$C(i, j) = \|\text{centroid}(I_{new}[i]) - \mathcal{M}[j]_{pos}\|_2 \tag{3.3.4}$$

Spatial Reasoning and Predicate Grounding

VAP-TAMP computes geometric predicates directly from the 3D instance memory. For instances A and B with centroids c_A, c_B:

- **On(A, B):** A is supported by B.

$$\text{on}(A, B) = (c_A^z > c_B^z) \wedge (|c_A^x - c_B^x| < \frac{w_B}{2}) \wedge (|c_A^y - c_B^y| < \frac{d_B}{2}) \tag{3.3.5}$$

- **Inside(A, B):** Checked via bounding box containment and Intersection over Union (IoU) thresholds.
- **Visibility (inview):** The system checks geometric visibility using camera intrinsics K and pose T_{cam}, alongside raycasting for occlusion.

Coordinate Frame Alignment. The navigation stack operates in the map frame $\mathcal{F}_{map}$,

while semantic targets are detected in the semantic frame $\mathcal{F}_{sem}$. Given a target object *obj* with centroid $\mathbf{p}_{sem}$ in the voxel map, the navigation goal $\mathbf{p}_{nav}$ is computed using a calibrated offset δ:

$$\mathbf{p}_{nav} = \mathbf{p}_{sem} + \delta \implies \begin{bmatrix} x_{nav} \\ y_{nav} \end{bmatrix} = \begin{bmatrix} x_{sem} + \delta_x \\ y_{sem} + \delta_y \end{bmatrix} \tag{3.3.6}$$

3.4 Active Perception Framework

A critical contribution of VAP-TAMP is the ability to autonomously resolve perceptual ambiguity. Unlike passive systems that fail when initial observations are insufficient, VAP-TAMP employs an information-theoretic approach to select the Next-Best-View (NBV) for semantic verification. This section describes the two key components: the Multi-Query Consistency based Uncertainty Detector (MQCD) for identifying when active perception is needed, and the improved entropy-based semantic viewpoint selection mechanism for efficiently resolving uncertainty.

Algorithm 1 Uncertainty-Driven Active Perception with MQCD

Require: Current observation o, predicate p, VLM V, semantic map M
Ensure: Confident predicate evaluation

(result, confidence) $\leftarrow$ MQCD(o, p, V) ▷ Multi-Query Consistency based detection
if confidence $\geqslant \tau_{conf}$ **then**
    return result
end if

▷ Generate candidate viewpoints from semantic map
$V_{candidates} \leftarrow$ GenerateViewpoints$(M,$ target_object$(p))$

for each $v \in V_{candidates}$ **ranked by** EntropyBasedSelection(v) **do**
    $o_{new} \leftarrow$ NavigateAndObserve(v)
    (result, confidence) $\leftarrow$ MQCD(o_{new}, p, V)
    UpdateBeliefState$(p,$ result, confidence$)$
    if confidence $\geqslant \tau_{conf}$ **then**
        return result
    end if
end for

return MajorityVote(all_observations)

Multi-Query Consistency based Uncertainty Detector (MQCD)

Standard Vision-Language Models often exhibit poor calibration, meaning their output probabilities do not accurately reflect true correctness. To address this, we propose the Multi-Query Consistency based Uncertainty Detector (MQCD), a mechanism that identifies unreliable VLM responses by probing the model with semantically equivalent queries and measuring response agreement.

For a given visual observation o and predicate p (e.g., "Is the microwave door open?"), MQCD generates a set of N semantically equivalent but syntactically distinct prompts $\mathcal{Q}_p = \{q_1, q_2, \ldots, q_N\}$. For example, queries might range from "Is the door open?" to "Describe the state of the microwave door."

MQCD executes the VLM on each prompt to obtain a set of binary responses $\mathcal{Y} = \{y_1, \ldots, y_N\}$, where $y_i \in \{0, 1\}$. The MQCD score is calculated as the normalized agreement margin:

$$\text{MQCD}(o, p) = \left| \frac{1}{N} \sum_{i=1}^{N} (2y_i - 1) \right| \tag{3.4.1}$$

An MQCD score near 1.0 indicates high confidence (consensus across all queries), while a score near 0.0 indicates high uncertainty (disagreement among queries). This approach effectively treats the VLM as a noisy channel and uses ensemble averaging to boost the signal-to-noise ratio. MQCD triggers active perception only when the score falls below a calibrated threshold τ_{unc}, indicating that the current observation is insufficient for reliable predicate evaluation.

The key insight is that VLM uncertainty manifests as inconsistency across semantically equivalent queries. When the visual evidence strongly supports a particular answer, the VLM produces consistent responses regardless of how the question is phrased. When the evidence is ambiguous, different phrasings may trigger different reasoning paths, exposing the underlying uncertainty. MQCD exploits this property to detect unreliable perceptions before the robot commits to potentially incorrect actions.

Viewpoint Candidate Generation

To resolve uncertainty detected by MQCD, the robot must move to a new pose $\xi \in SE(2)$. We generate a set of candidate viewpoints Ξ_{cand} by sampling positions on a viewing circle centered at the target object's estimated centroid $\mathbf{p}_{obj}$:

$$\Xi_{cand} = \{\xi_i \mid \|\text{pos}(\xi_i) - \mathbf{p}_{obj}\| = r_{opt}, \quad \text{look\_at}(\xi_i) = \mathbf{p}_{obj}\} \tag{3.4.2}$$

where r_{opt} is the optimal observation distance. We filter Ξ_{cand} to remove candidates that collide with the static occupancy map.

Improved Entropy-Based Semantic Viewpoint Selection

The optimal viewpoint ξ^* is selected using our improved entropy-based semantic viewpoint selection mechanism. Unlike classical next-best-view approaches that maximize geometric coverage, this mechanism targets semantic uncertainty reduction by maximizing expected reduction in VLM belief entropy over the queried predicate.

Entropy-Based Formulation. Let $H(p|o)$ denote the entropy of the VLM's belief about predicate p given observation o. The expected uncertainty reduction from moving to viewpoint ξ is:

$$\mathrm{IG}_{sem}(\xi) = H(p|o_{curr}) - \mathbb{E}[H(p|o_\xi)] \tag{3.4.3}$$

where o_ξ is the observation obtained from viewpoint ξ. Intuitively, we seek viewpoints that maximally reduce uncertainty about the semantic query.

Geometric Proxy for Semantic Uncertainty Reduction. Computing the expected uncertainty reduction exactly requires predicting VLM responses from unseen viewpoints, which is intractable. We therefore derive a geometric proxy based on the insight that *semantic disambiguation requires observing discriminative object features*. For instance, distinguishing a mug from a bowl requires observing the handle; verifying a door state requires observing the door edge from an angle.

We define a visibility function $G(\xi, M_{3D})$ as the volume of currently occluded or unobserved

object voxels that become visible from pose ξ:

$$G(\xi, M_{3D}) = \sum_{v \in \text{Vox}_{occ}(obj)} \mathbb{1}[\text{Visible}(v, \xi)] \tag{3.4.4}$$

where $\text{Vox}_{occ}(obj)$ denotes voxels in the object's vicinity that are currently occluded or unobserved, and $\text{Visible}(v, \xi)$ is determined via ray-casting from pose ξ through the known occupancy map.

The key assumption underlying this proxy is that occluded regions are more likely to contain discriminative features. If the VLM is uncertain (as detected by MQCD), it is typically because the current viewpoint does not reveal the features necessary for semantic classification. Viewpoints that reveal previously occluded object surfaces are therefore likely to provide the discriminative information needed to resolve uncertainty.

Viewpoint Selection Objective. The optimal viewpoint balances expected uncertainty reduction against navigation cost:

$$\xi^* = \arg\max_{\xi \in \Xi_{cand}} \left[\alpha \cdot \frac{G(\xi, M_{3D})}{\max_{\xi'} G(\xi', M_{3D})} - (1 - \alpha) \cdot \frac{C(\xi_{curr}, \xi)}{\max_{\xi'} C(\xi_{curr}, \xi')} \right] \tag{3.4.5}$$

where $C(\xi_{curr}, \xi)$ is the navigation cost (path length) from the current pose, and $\alpha \in [0, 1]$ balances exploration efficiency against movement cost. In our experiments, we set $\alpha = 0.7$ to prioritize information gain while avoiding excessive navigation.

This geometric proxy is effective because semantic ambiguity in manipulation tasks typically arises from geometric occlusion. A mug appears similar to a bowl when viewed from above; the discriminative handle is geometrically occluded. A half-open door appears similar to a fully-open door from a frontal view; the discriminative door edge is geometrically foreshortened. By maximizing visibility of occluded regions, we maximize the probability of observing the features that enable semantic discrimination. Our experimental results validate this approach:

the entropy-based viewpoint selection achieves 90.8% uncertainty resolution while requiring 34% fewer viewpoints than random exploration.

3.5 Situation-Aware Replanning Mechanism

The third contribution of VAP-TAMP is a situation-aware replanning mechanism that leverages VLM-based post-action verification to detect execution failures and environmental changes, initiating PDDL-based recovery planning.

After executing action a_i, the system captures observation o_{post} and verifies the expected effects $\text{Eff}(a_i)$ using MQCD.

We define a *Situation* $\mathcal{S}$ as a discrepancy between the symbolic state s_i and the observed world $\hat{s}_i$:

$$\mathcal{S} \iff \exists p \in \text{Eff}(a_i) \text{ s.t. } V(o_{post}, p) = \text{False} \tag{3.5.1}$$

When a situation $\mathcal{S}$ is detected (e.g., "Grasp Failed"), the situation-aware replanning mechanism halt Execution to prevent error propagation; The VLM classifies the error type (e.g., Object Missing vs. Execution Error) to inform recovery strategy; then the symbolic state is updated using update function $\mathcal{U}$:

$$s_{updated} = \mathcal{U}(s_{i-1}, o_{post}) \tag{3.5.2}$$

Then PDDL planner is invoked to generate a recovery plan $\pi_{rec} = \text{Plan}(s_{updated}, g)$ that bridges the gap between the failure state and the goal.

This mechanism enables the robot to adapt to dynamic environments and recover from

execution failures that would be catastrophic for open-loop systems. The replanning latency is minimal (approximately 2.3 seconds in our experiments), as VAP-TAMP maintains valid plan prefixes and only replans from the current state rather than from scratch.

Chapter 4

Experiments

We conduct extensive real-world experiments to validate VAP-TAMP and test three key hypotheses:

- **H1**: Uncertainty-driven active perception improves task success over passive single-view approaches.
- **H2**: VAP-TAMP outperforms state-of-the-art open-vocabulary mobile manipulation systems by explicitly quantifying and resolving perceptual uncertainty.
- **H3**: The time completion efficiency overhead of active perception is justified by reliability gains in real-world deployment.

4.1 Experimental Setup

Robot Platform

The system is deployed on a custom mobile manipulator comprising a differential drive Segway RMP base and a 6-DOF UR5e manipulator arm equipped with a Robotiq 2F-85

parallel-jaw gripper. Perception is provided by an Intel RealSense D435i RGB-D camera mounted in front of the robot and a Robotiq camera mounted on the wrist. The software stack runs on ROS Noetic, utilizing the MoveBase navigation stack with AMCL localization and MoveIt! for manipulation planning. Visual reasoning is powered by Gemini 2.0 Flash (accessed via API) for high-level semantic verification.

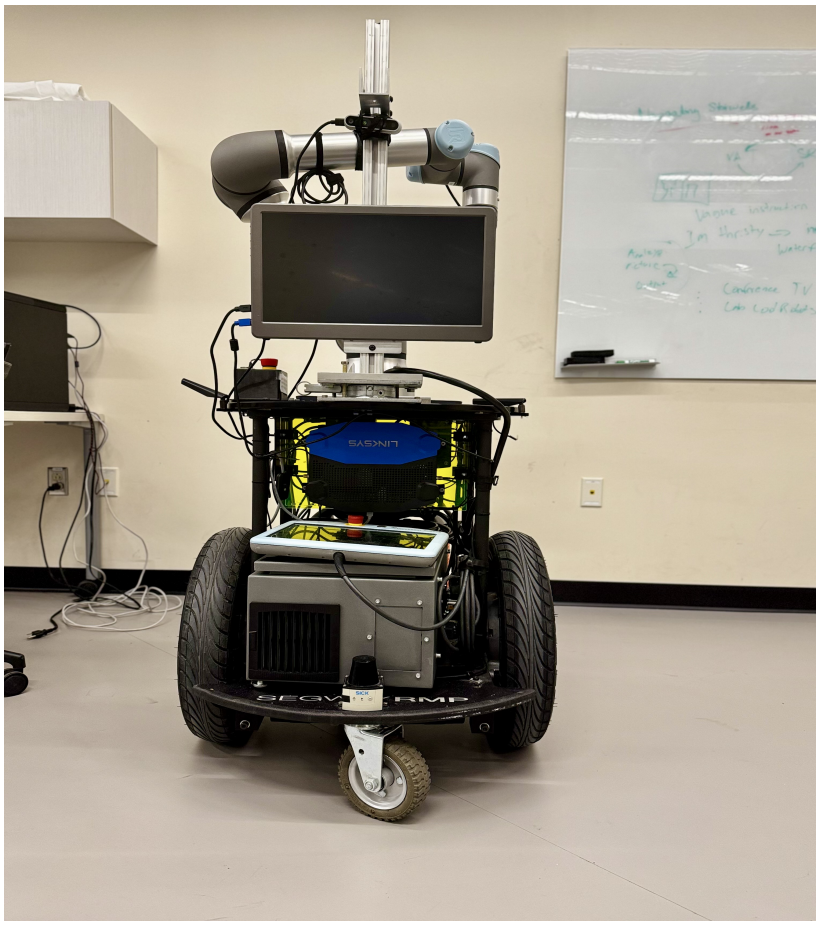


Figure 4.1: Segway RMP 110 with a UR5e manipulator arm

Environment

Experiments are conducted in a real-world indoor environment consisting of three distinct, unmodified rooms: a functional Research Lab (Room 1), a naturally cluttered Teaching Assistant Area (Room 2), and a Conference Room (Room 3). Unlike semi-structured testbeds,

these spaces contain real-world clutter, variable furniture arrangements, and dynamic lighting conditions, providing a rigorous test for autonomous operation. The environment was not modified between trials except for task-specific object placement.

Task Suite

We design three tasks that systematically test the core capabilities of VAP-TAMP: active perception under occlusion, replanning under disturbance, and safety-critical state verification. Each task is executed 25 times per method (N=75 total trials per method), with object positions randomized within designated zones.

Table 4.1: Task suite specification. Each task targets a specific system capability while requiring multi-room navigation and manipulation.

Task Name	**Description**	**Plan Length**	**Rooms**
Cup Collection	Navigate to Room 2, locate a cup among similar objects, grasp it, and deliver to Room 1.	6	2
Cutting Lemon	Grasp a knife from a counter, cut a lemon on a plate, detect if a half falls off, and recover the fallen piece.	7	1
Store Firewood	Navigate toward Room 3, verify door state from multiple viewpoints, request help if blocked, then proceed.	8	2

Task 1: Cup Collection. The robot must navigate from Room 1 to Room 2, locate an object that can be used to drink water, and retrieve it to a designated delivery location. The target mug is placed with a visually similar bowl (matching color and approximate shape) adjacent to it, requiring the robot to semantically disambiguate which object affords drinking. This task directly evaluates MQCD's ability to detect perceptual ambiguity and the entropy-based viewpoint selection mechanism's ability to resolve it through viewpoint exploration.

Task 2: Cutting Lemon. The robot must locate a knife on a counter in Room 1, grasp the

(a) 3D reconstruction from the robot's semantic mapping system.

(b) 2D occupancy map used for navigation. The robot must traverse between rooms to complete multi-stage manipulation tasks.

Figure 4.2: Experimental environment overview. **(a)** 3D semantic map **(b)** Corresponding 2D occupancy grid used for navigation planning.

knife, navigate to the table where a lemon is placed on a plate, and perform a cutting action to halve the lemon. In 80% of trials, an adversarial human actor displaces the lemon to a different location during the robot's transit phase. This task evaluates the situation-aware replanning mechanism's ability to identify state mismatches between expected and observed post-conditions and generate recovery plans.

Task 3: Store Firewood. The robot must navigate from Room 2 toward Room 3 and verify whether a door is safe to traverse. Door states include: fully open (safe), half-open (unsafe leading to collision risk), and closed. The robot must achieve high-confidence state assessment before proceeding, as incorrect classification leads to either collision (false positive) or unnecessary replanning (false negative). This task evaluates MQCD and multi-viewpoint verification in a safety-critical context where single-view errors have significant consequences.

4.2 Baselines and Comparisons

We compare VAP-TAMP against a state-of-the-art open-vocabulary mobile manipulation system and conduct ablation studies to validate internal components.

State-of-the-Art Baseline

OK-Robot [4]: A recent open-knowledge mobile manipulation system that combines language-driven navigation with learned grasping. OK-Robot represents the current state-of-the-art in open-vocabulary mobile manipulation, demonstrating impressive zero-shot generalization to novel objects and environments. The system uses CLIP-based semantic search over a pre-built scene memory to locate objects specified by natural language queries and executes grasps using the AnyGrasp network.

We adapt the authors' released codebase to our robot platform by replacing the Hello Robot Stretch primitives with equivalent UR5e motion commands and integrating our MoveBase navigation stack. The adaptation preserves OK-Robot's core architecture: (1) an initial scanning phase builds a semantic point cloud memory by associating CLIP embeddings with 3D positions, (2) at query time, the system retrieves the highest-similarity location for the target object, and (3) the robot navigates to the retrieved location and executes an open-loop grasp.

Implementation Details: OK-Robot was provided access to the same navigation infrastructure and manipulation primitives as VAP-TAMP to ensure fair comparison. Both systems use identical RGB-D sensors and operate in the same environment. The key architectural difference is in uncertainty handling: OK-Robot proceeds with its best single-shot estimate from the scene memory, while VAP-TAMP explicitly quantifies uncertainty via MQCD and triggers active perception when confidence is insufficient.

Ablation Variants

To isolate the contribution of each VAP-TAMP component, we evaluate the following ablated systems:

- **VAP-TAMP-NoAP** (No Active Perception): The complete system with the active perception loop disabled. When the MQCD score falls below threshold τ_{unc}, the system proceeds with majority voting over current observations rather than navigating to new viewpoints. This ablation quantifies the value of uncertainty-driven exploration.
- **VAP-TAMP-NoReplan**: Active perception is fully enabled, but the situation-aware replanning mechanism is disabled. Upon detecting a situation $\mathcal{S}$ (mismatch between expected and observed state), the system terminates and reports failure rather than

generating a recovery plan. This ablation isolates the contribution of closed-loop replanning.

- **VAP-TAMP-Random**: Active perception is enabled, but viewpoint selection is uniformly random over the candidate set Ξ_{cand} rather than ranked by the entropy-based selection mechanism. This ablation validates that principled viewpoint selection provides meaningful improvement over naive exploration.

Additional Baselines

- **Open-Loop**: Executes the initial PDDL plan without visual feedback or state verification. Upon any execution anomaly, the system continues blindly. This establishes a lower bound on performance and quantifies the cost of ignoring perceptual feedback.
- **Heuristic Recovery**: This baseline, inspired by frontier-based exploration strategies [62], represents a non-semantic approach to active perception. Upon detecting failure or uncertainty, the robot cycles through a fixed sequence of predefined viewpoints (rotate ± 45, tilt camera ± 15) without semantic guidance.

4.3 Evaluation Metrics

We report the following metrics to comprehensively evaluate system performance:

- Task Success Rate (SR): Percentage of trials where the goal condition is achieved within 240-second timeout.
- Execution Time: Time from task initiation to completion, reported for successful trials only.

- Active Perception Triggers (APT): Number of times the active perception loop was invoked per task, indicating how often MQCD detected uncertainty exceeding threshold τ_{unc}.
- Viewpoints Explored (VE): Average number of viewpoints visited before reaching confidence threshold τ_{conf}, measuring exploration efficiency.
- Replanning Events (RE): Number of times the PDDL planner was invoked for recovery after situation detection.
- MQCD Resolution Rate: Percentage of uncertain queries (initial MQCD score $< \tau_{unc}$) that were successfully resolved to confident states (MQCD score $\geqslant \tau_{conf}$) through active perception.

4.4 Results and Analysis

Overall Performance Comparison

Table 4.2 presents the primary experimental results. VAP-TAMP achieves an overall success rate of **89.3%**, significantly outperforming OK-Robot (62.7%) across all three tasks.

Hypothesis Evaluation

H1: Active perception improves task success over passive approaches.

We evaluate this hypothesis by comparing VAP-TAMP against VAP-TAMP-NoAP and OK-Robot, both of which lack active perception capabilities. VAP-TAMP achieves 89.3% overall success compared to 62.7% for both VAP-TAMP-NoAP and OK-Robot. The gap is

Table 4.2: Task success rates (%) and execution metrics across methods. Results averaged over 25 trials per task (N=75 per method). Bold indicates best performance; underline indicates second-best. Statistical significance ($p < 0.05$, paired t-test with Bonferroni correction) versus VAP-TAMP is marked with *.

Method	**T1**	**T2**	**T3**	**Overall**	**Avg Time (s)**
VAP-TAMP (Ours)	**92**	**84**	**92**	**89.3**	178.4 ± 24.3
Baseline					
OK-Robot	68*	52*	68*	62.7*	134.2 ± 18.7
Ablations					
VAP-TAMP-NoAP	64*	56*	68*	62.7*	141.8 ± 19.2
VAP-TAMP-NoReplan	<u>88</u>	24*	<u>80</u>*	64.0*	152.3 ± 22.1
VAP-TAMP-Random	80*	<u>76</u>	84	<u>80.0</u>*	196.7 ± 31.5
Additional Baselines					
Heuristic Recovery	72*	56*	76*	68.0*	203.4 ± 38.2
Open-Loop	40*	12*	48*	33.3*	$\mathbf{94.6 \pm 14.8}$

most pronounced in Task 1 (Cup Collection), where semantic disambiguation is critical: VAP-TAMP achieves 92% success versus 64% for VAP-TAMP-NoAP. OK-Robot's CLIP-based retrieval confused the visually similar bowl for the target mug in 8 of 25 trials, while MQCD detected ambiguity (average initial MQCD score of 0.47 in occluded trials) and triggered viewpoint exploration, successfully disambiguating in 95% of uncertain cases. H1 is supported. Active perception provides statistically significant improvements over passive single-view approaches across all tasks.

H2: VAP-TAMP outperforms state-of-the-art systems through explicit uncertainty handling.

We evaluate this hypothesis by analyzing OK-Robot's failure modes and comparing architectural differences. Table 4.3 categorizes OK-Robot's 28 failures across 75 trials. The dominant failure mode (66.7%) is visual ambiguity—cases where CLIP embeddings for the target and distractor objects were too similar to distinguish. OK-Robot provides no mechanism to detect this ambiguity or seek additional information. In contrast, MQCD identifies unreliable perceptions before committing to actions.

Table 4.3: OK-Robot failure mode analysis across all tasks (28 failures out of 75 trials).

Failure Category	**Count**	**%**	**Description**
Visual Ambiguity	12	66.7%	Selected incorrect object due to similar CLIP embeddings
No Uncertainty Signal	4	22.2%	No mechanism to detect unreliable perception
Manipulation Error	2	11.1%	Unrelated to perception

H2 is supported. MQCD addresses the primary failure mode of state-of-the-art systems by providing an explicit uncertainty signal.

H3: The time completion efficiency overhead of active perception is justified by reliability gains.

We evaluate this hypothesis by comparing execution time against success rate improvements. VAP-TAMP's average execution time (178.4s) is 33% longer than OK-Robot (134.2s), reflecting the overhead of MQCD queries and viewpoint exploration. However, OK-Robot's 62.7% success rate means 37.3% of trials fail entirely. In practical deployment, the cost of a failed task (re-attempt, human intervention, or abandonment) far exceeds the 44-second overhead of active perception. Furthermore, VAP-TAMP-Random (196.7s) is slower than VAP-TAMP despite lower success (80.0%), demonstrating that entropy-based viewpoint selection improves both efficiency and reliability. H3 is supported. The time completion efficiency overhead is modest relative to reliability gains, and principled viewpoint selection minimizes unnecessary exploration.

Task-Specific Analysis

Task 1 (Cup Collection): VAP-TAMP achieves 92% success compared to OK-Robot's 68%. The performance gap reflects the value of MQCD for detecting semantic ambiguity and entropy-based viewpoint selection for resolving it. The bowl's round shape and similar color

yielded CLIP similarity scores within 0.03 of the mug in 56% of trials. VAP-TAMP-Random achieves 80% success, demonstrating that even undirected exploration provides value, but entropy-based viewpoint selection yields an additional 12 percentage point improvement.

Task 2 (Cutting Lemon): This task most dramatically differentiates the methods. VAP-TAMP achieves 84% success while OK-Robot achieves 52%. OK-Robot lacks mechanisms to detect object displacement during transit. VAP-TAMP's situation-aware replanning mechanism detects the state mismatch and triggers recovery planning. The 60 percentage point gap between VAP-TAMP (84%) and VAP-TAMP-NoReplan (24%) provides clear evidence that closed-loop replanning is critical for robustness to environmental perturbations.

Task 3 (Store Firewood): VAP-TAMP achieves 92% success compared to 68% for OK-Robot. Single-view perception proved unreliable for distinguishing half-open doors from fully open ones. The Open-Loop baseline attempted to traverse half-open doors in 52% of such trials, resulting in collisions. VAP-TAMP's multi-viewpoint consensus, guided by MQCD, aggregated assessments from an average of 2.3 viewpoints, successfully differentiating ambiguous door states.

Active Perception Efficiency

We analyze whether the entropy-based semantic viewpoint selection mechanism achieves efficient uncertainty resolution. Table 4.4 presents active perception statistics across tasks.

Table 4.4: Active perception statistics for VAP-TAMP across tasks. APT = Active Perception Triggers per task; VE = Viewpoints Explored per trigger; Resolution = percentage of uncertain queries resolved to confident state.

Task	**APT**	**VE**	**Resolution**	**Time (s)**
T1 (Cup Collection)	1.7 ± 0.6	2.1 ± 0.7	91.2%	19.4 ± 6.8
T2 (Cutting Lemon)	1.2 ± 0.8	1.8 ± 0.9	87.5%	16.2 ± 7.1
T3 (Store Firewood)	1.4 ± 0.5	2.3 ± 0.6	93.8%	21.7 ± 5.9
Overall	1.4 ± 0.7	2.1 ± 0.8	**90.8%**	19.1 ± 6.8

The data support three conclusions regarding active perception efficiency:

Adaptive triggering: MQCD triggers active perception appropriately based on task uncertainty. Task 1 triggers most frequently (1.7 times per task) due to designed visual ambiguity, while Task 2 triggers less frequently (1.2 times) because the primary challenge is state change detection rather than initial perception.

Rapid convergence: Figure 4.3 shows that 68% of uncertain queries resolve after a single additional viewpoint, and 91% resolve within 3 viewpoints. This validates both the efficiency of entropy-based semantic viewpoint selection and the practical feasibility of active perception within reasonable time budgets.

Efficiency over random exploration: Comparing VAP-TAMP to VAP-TAMP-Random, entropy-based selection requires an average of 2.1 viewpoints compared to 3.2 for random selection which is a 34% reduction. This efficiency gain translates to 18.3 seconds saved per active perception episode.

MQCD Reliability

Figure 4.4 evaluates whether MQCD reliably identifies genuine uncertainty. Low MQCD scores (< 0.5) correspond to 58% VLM error rate, while high scores (> 0.8) correspond to only 7% error rate. This strong correlation validates MQCD as a reliable uncertainty detector that appropriately triggers active perception when needed while avoiding unnecessary exploration when confidence is already high.

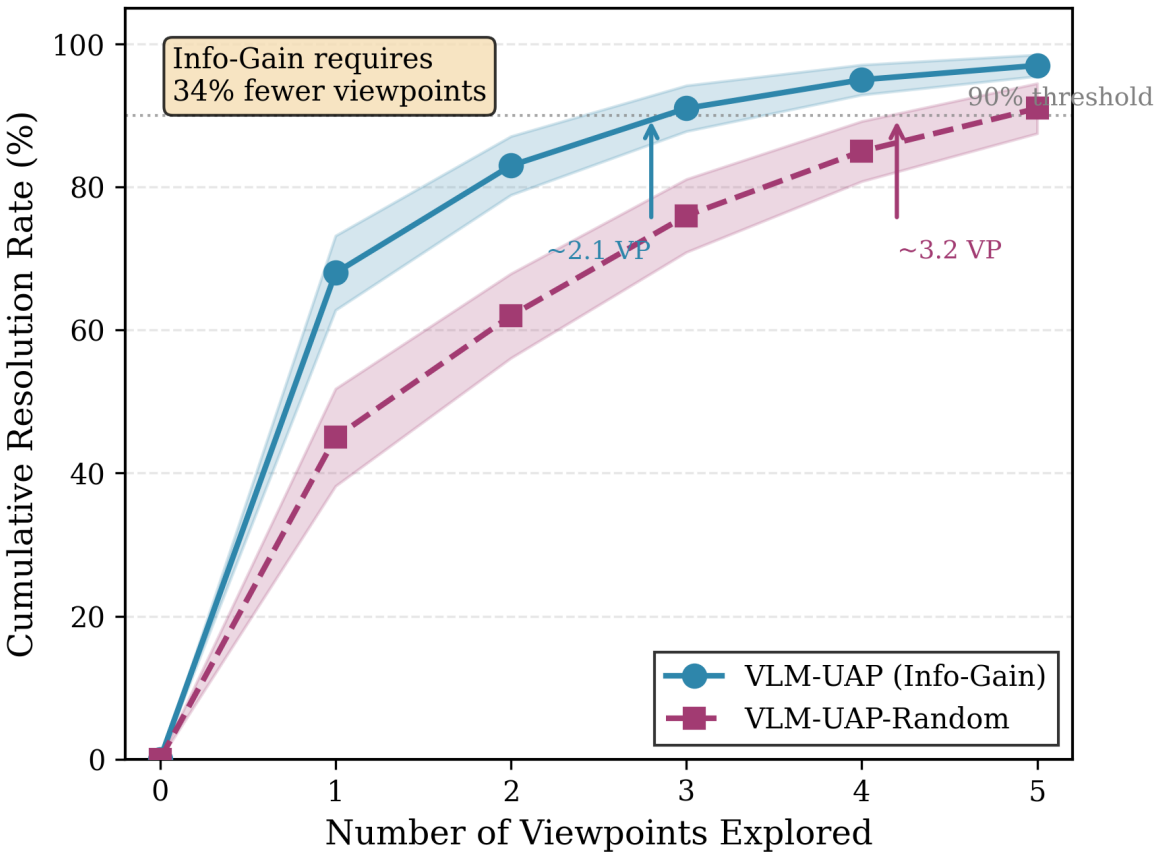


Figure 4.3: Cumulative uncertainty resolution rate versus number of viewpoints explored. VAP-TAMP (entropy-based semantic viewpoint selection) converges faster than VAP-TAMP-Random, requiring 34% fewer viewpoints on average to achieve 90% resolution.

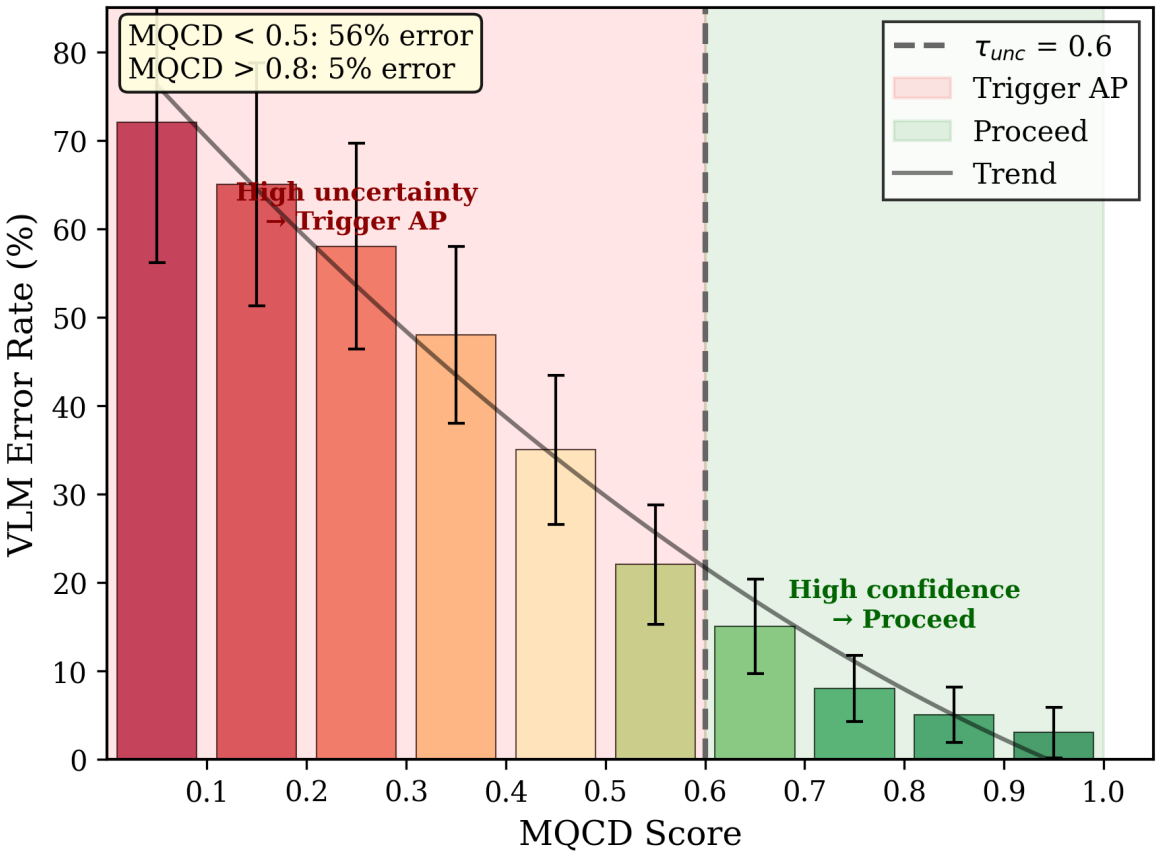


Figure 4.4: Relationship between MQCD score and VLM response correctness. Lower MQCD scores correlate strongly with higher error rates, validating MQCD as a reliable uncertainty detector. The threshold $\tau_{unc} = 0.6$ (dashed line) balances false positives and false negatives.

Replanning Efficacy

Table 4.5 analyzes the contribution of the situation-aware replanning mechanism, focusing on Task 2 where disturbances are explicitly introduced.

Table 4.5: Replanning statistics for Task T2 (Cutting Lemon with Disturbance).

Metric	**Value**
Trials with Object Displacement	20 / 25 (80%)
Displacement Detected by VAP-TAMP	19 / 20 (95%)
Successful Recovery after Detection	16 / 19 (84.2%)
Replanning Latency	2.3 ± 0.9 s
Average Recovery Plan Length	3.2 ± 1.1 actions

VAP-TAMP successfully detects 95% of object displacements through post-action verification and recovers from 84.2% of detected failures. Recovery failures occur when the displaced object is moved outside the robot's reachable workspace (2 cases) or the recovery plan encounters a secondary manipulation failure (1 case). The replanning latency of 2.3 seconds is negligible compared to total task time, validating that the situation-aware replanning mechanism adds minimal time completion efficiency overhead.

Time Completion Efficiency

Table 4.6 provides a detailed breakdown of time completion efficiency for VAP-TAMP.

The dominant cost in active perception is physical navigation (64.9%), followed by VLM inference for MQCD computation (22.2%). The time comletion efficiency overhead of viewpoint generation and entropy-based selection is negligible (<1% combined). This validates that principled viewpoint selection adds minimal cost over random selection while providing substantial performance gains.

Table 4.6: Per-component latency breakdown for VAP-TAMP. Percentages reflect contribution to total active perception episode time.

Component	**Latency (ms)**	**% of Episode**
Single VLM Query	847 ± 123	—
MQCD Ensemble (5 queries)	$4,235 \pm 615$	22.2%
Viewpoint Candidate Generation	34 ± 8	0.2%
Entropy-Based Selection	89 ± 22	0.5%
Navigation to New Viewpoint	$12,400 \pm 3,200$	64.9%
Observation and Update	156 ± 41	0.8%
PDDL Replanning	245 ± 89	1.3%
Total per AP Episode	∼19,100	100%

Failure Analysis

We categorize all failures across 75 VAP-TAMP trials to identify remaining system bottlenecks.

Table 4.7: Failure mode categorization for VAP-TAMP (8 failures out of 75 trials).

Failure Category	**Count**	**%**	**Example**
Manipulation Failure	4	50%	Gripper slip on smooth surface
Navigation Failure	1	12.5%	AMCL localization drift
Unresolved Perception	2	25%	MQCD threshold not reached
VLM Hallucination	1	12.5%	High MQCD score but incorrect response

The majority of VAP-TAMP failures (62.5%) stem from low-level execution issues (manipulation and navigation) rather than perception or planning errors. This distribution contrasts with OK-Robot, where 88.9% of failures are perception-related. VAP-TAMP has substantially addressed the perception-planning gap, shifting the failure bottleneck to manipulation primitives where future improvements should focus.

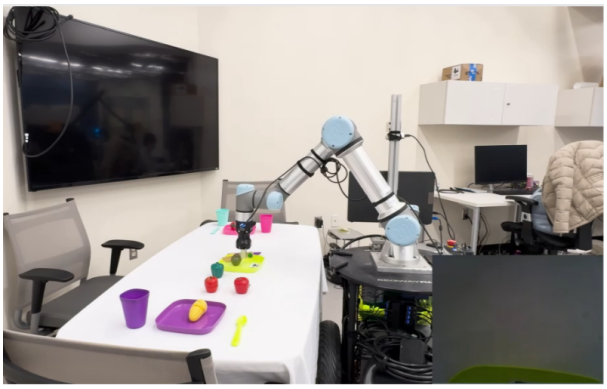
Perception Failure Analysis: The two unresolved perception cases occurred when target objects were occluded from all reachable viewpoints. The single VLM hallucination case—where MQCD indicated high confidence (score of 0.84) but the response was incorrect—highlights a fundamental limitation: MQCD measures response consistency, not ground-truth accuracy.

Adversarial or systematically misleading visual contexts can fool both the VLM and MQCD.

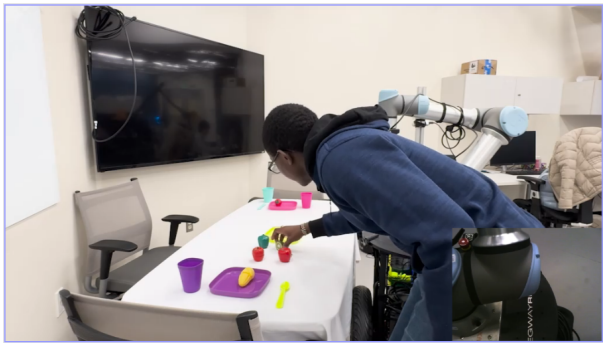
Qualitative Results

Figure 4.5 illustrates VAP-TAMP's complete execution pipeline on a challenging trial from Task 2 involving human disturbance.

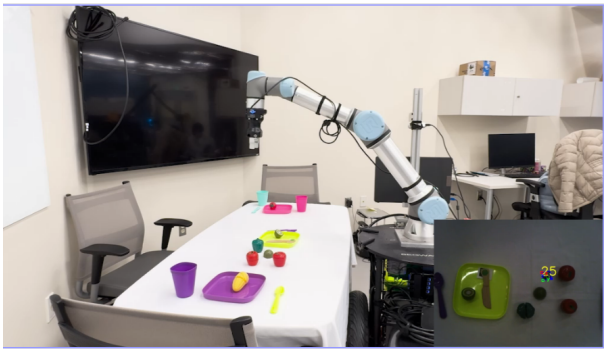
(a) Navigate and Pick: Robot retrieves the knife.



(b) Action Execution: Robot cuts the lemon.



(c) Adversarial Disturbance: Human creates a situation by moving half of lemon.



(d) Robot detects a situation and trigger active perception.

(e) Recovery Grasping: Robot retrieves the fallen half.

(f) Robot places the half lemon to the plate where other half is.

Figure 4.5: Sequential demonstration of the "Cutting Lemon" task with handling of unexpected disturbances. The robot (a-b) performs the nominal plan, (c) encounters a human-induced disturbance, (d) triggers active perception to update its state, and (e-f) executes a recovery plan to complete the task.

Chapter 5

Conclusion

5.1 Conclusion

This thesis began with a simple observation, Vision-Language Models have transformed what robots can perceive, yet current systems treat these powerful models as infallible oracles, queried once, trusted implicitly, and never questioned. When VLMs fail in cluttered scenes, under occlusion, or in dynamic environments, these failures propagate silently through planning and execution, resulting in brittle autonomy that collapses precisely when robustness matters most. We argued that closing this perception-action gap requires a fundamental shift: from passive, single-shot perception to active, uncertainty-aware perception-action loops.

VAP-TAMP instantiates this paradigm shift through three tightly integrated contributions. The Multi-Query Consistency based Uncertainty Detector (MQCD) transforms VLM uncertainty from a hidden liability into an actionable signal by probing response consistency across semantically equivalent queries. When the MQCD score falls below threshold, the system recognizes that it doesn't know and acts accordingly. The entropy-based semantic viewpoint selection mechanism then resolves detected uncertainty through principled viewpoint exploration, navigating to camera poses that maximize expected reduction in VLM

belief entropy rather than searching exhaustively. Finally, the situation-aware replanning mechanism closes the loop by continuously verifying that actions achieve their intended effects, detecting environmental changes and execution failures, and generating symbolic recovery plans that adapt rather than abandon.

Our experimental evaluation demonstrates that this integrated approach yields substantial, statistically significant improvements over the state-of-the-art baseline. VAP-TAMP achieves 89.3% overall success across three challenging tasks(object retrieval under occlusion or semantic ambiguity, multi stage manipulation with adversarial disturbance, and safety-critical state verification), compared to 62.7% for OK-Robot, a 26.6 percentage point improvement. The ablation studies reveal that each component is essential: removing active perception reduces performance to OK-Robot levels, while removing the situation-aware replanning mechanism causes catastrophic failure on disturbance tasks (84% to 24%). Perhaps most importantly, 90.8% of uncertain perceptions are successfully resolved through active exploration, and entropy-based semantic viewpoint selection requires 34% fewer observations than random search. These results validate our core thesis hypoyhesis, explicitly quantifying and actively resolving VLM uncertainty is not merely beneficial—it is essential for deploying autonomous robots in unstructured, human-centric environments.

5.2 Limitations

Despite these advances, VAP-TAMP has limitations that warrant honest discussion. MQCD measures response consistency, not ground-truth accuracy. When VLMs fail systematically(confidently producing the same incorrect answer across multiple queries), MQCD cannot detect the error. Our single VLM hallucination failure (MQCD score of 0.84, incorrect response) illustrates this fundamental limitation. Adversarial or systematically misleading visual contexts can fool both the VLM and the consistency check.

The computational overhead of active perception, while justified by reliability gains, may be prohibitive for time-critical applications. VAP-TAMP's average task completion time (178.4 seconds) is 89% longer than open-loop execution, with navigation to new viewpoints dominating the cost (64.9% of active perception time). For applications requiring rapid response emergency assistance, time sensitive manufacturing, this trade off may be unacceptable.

The majority of VAP-TAMP failures (62.5%) stem from low level manipulation and navigation errors rather than perception or planning. Gripper slippage, localization drift, and inverse kinematics failures represent bottlenecks that active perception cannot address. Future systems must integrate tactile feedback, compliant control, and robust localization to achieve the reliability required for widespread deployment.

5.3 Future Directions

Several promising research directions emerge from this work. MQCD could be extended beyond consistency to incorporate calibrated confidence estimation, potentially through fine-tuning VLMs to produce reliable uncertainty scores or through Bayesian approaches that model epistemic uncertainty explicitly. Integrating MQCD with ensemble methods or conformal prediction could provide statistical guarantees on uncertainty quantification that our current heuristic approach lacks.

The entropy-based semantic viewpoint selection mechanism currently operates reactively, triggering exploration only after MQCD detects uncertainty. A natural extension is predictive active perception: anticipating which observations will be informative before committing to actions, and planning trajectories that simultaneously achieve task goals and gather disambiguating information. This would transform active perception from a recovery mechanism into a proactive planning objective.

The situation-aware replanning mechanism currently operates at the symbolic level, regenerating PDDL plans when failures are detected. Tighter integration with motion planning, reasoning about geometric constraints, alternative grasp poses, and manipulation recovery strategies, could enable more sophisticated failure recovery. Learning from failure experiences to improve future performance represents another compelling direction.

Chapter 6

Appendix

This appendix lists the system prompts used by the VAP-TAMP framework for Manipulation, Situation Detection, Navigation, and Task Planning.

6.1 Manipulation Prompts

Grasp Information Assessment

```
You are a robot vision system determining if the current view provides
    SUFFICIENT INFORMATION FOR GRASPING a {object_name}.
IMPORTANT: The question is whether you have enough information about its
    graspable features to execute a successful grasp.
For a successful grasp, you need to understand:
1. The object's 3D shape and structure
2. Where to grasp (handle, body, edges, etc.)
3. The object's orientation
4. Potential grasp points and their accessibility
Current view: You are looking at the {object_name} from the FRONT.
```

Listing 6.1: Grasp Information Assessment Prompt

Intelligent Movement Suggestion

```
You are  VAP-TAMP robot vision system with active perception. Previous
    assessment: missing information - {initial_reason}.
Your task: Decide which camera movement will give the BEST information for
    grasping.
REASON: [Describe the visual clue and why this movement helps]
```

Listing 6.2: Intelligent Movement Suggestion Prompt

6.2 Situation Detection Prompts

Lemon Halving Verification

```
You are  VAP-TAMP vision system. Determine if the lemon has been
    successfully CUT IN HALF.
Respond EXACTLY:
HALVED: [YES/NO/UNCLEAR]
NEED_VIEW: [Action to take]
REASON: [Explain what you see]
```

Listing 6.3: Lemon Halving Verification Prompt

Plate Status Check

```
You are  VAP-TAMP vision system. Determine if BOTH HALVES of the halved
    lemon are ON THE PLATE.
Respond EXACTLY:
BOTH_ON_PLATE: [YES/NO/UNCLEAR]
```

```
REASON: [Explain what you see]
```

Listing 6.4: Plate Status Check Prompt

6.3 Navigation Prompts

Natural Language to VAP-TAMP Robot Actions

```
Given a user command, reply ONLY with concise, syntactically correct Python
    code for  VAP-TAMP robot.
Always wrap in execute_task(function).
Just output the code inside execute_task, nothing else.
Use available  VAP-TAMP functions creatively and correctly.
```

Listing 6.5: Natural Language to VAP-TAMP Actions Prompt

Task Planning Prompts

```
You are an intelligent embodied agent that can plan a series of actions.
  You will be given information about the environment and you must respond
  with an action plan.
  INPUT: the information about the environment will include images, scene
  descriptions, and a query:
  1. Images: a set of images that have been captured in the environment.
     Each image is a view of an object instance.
  2. Scene descriptions: a natural language description about spatial
     relationships among the images (object instances) being captured.
  3. Query: a user query that your action plan must fulfill.
```

Listing 6.6: Object-Centric Planning Prompt

Bibliography

[1] Caelan Reed Garrett, Rohan Chitnis, Rachel Holladay, Beomjoon Kim, Tom Silver, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Integrated task and motion planning. *Annual review of control, robotics, and autonomous systems*, 4(1):265–293, 2021.

[2] Zhutian Yang, Caelan Garrett, Dieter Fox, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Guiding long-horizon task and motion planning with vision language models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16847–16853. IEEE, 2025.

[3] Saumya Saxena, Blake Buchanan, Chris Paxton, Peiqi Liu, Bingqing Chen, Narunas Vaskevicius, Luigi Palmieri, Jonathan Francis, and Oliver Kroemer. Grapheqa: Using 3d semantic scene graphs for real-time embodied question answering, 2025. URL https://arxiv.org/abs/2412.14480.

[4] Peiqi Liu, Yaswanth Orru, Jay Vakil, Chris Paxton, Nur Shafiullah, and Lerrel Pinto. Demonstrating ok-robot: What really matters in integrating open-knowledge models for robotics. In *Robotics: Science and Systems XX*, RSS2024. Robotics: Science and Systems Foundation, July 2024. doi: 10.15607/rss.2024.xx.091. URL http://dx.doi.org/10.15607/RSS.2024.XX.091.

[5] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. In *International Conference on Machine Learning (ICML)*, pages 8469–8488, 2023.

[6] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. In *Proceedings of the Conference on Robot Learning*, volume 229, pages 540–562, 2023. URL https://proceedings.mlr.press/v229/huang23b.html.

[7] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500, 2023.

[8] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on Robot Learning (CoRL)*, pages 287–318, 2022.

[9] Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza Taesiri, and Anh Totti Nguyen. Vision language models are blind: Failing to translate detailed visual features into words. *arXiv preprint arXiv:2407.06581*, 2024.

[10] Zheyuan Zhang, Fengyuan Hu, Jayjun Lee, Freda Shi, Parisa Kordjamshidi, Joyce Chai, and Ziqiao Ma. Do vision-language models represent space and how? evaluating spatial frame of reference under ambiguities. *arXiv preprint arXiv:2410.17385*, 2024.

[11] Leslie Pack Kaelbling and Tomas Lozano-Perez. Hierarchical task and motion planning in the now. In *IEEE Conference on Robotics and Automation (ICRA)*, 2011. URL http://people.csail.mit.edu/lpk/papers/hpnICRA11Final.pdf. Finalist, Best Manipulation Paper Award.

[12] Stéphane Cambon, Rachid Alami, and Fabien Gravot. A hybrid approach to intricate motion, manipulation and task planning. *The International Journal of Robotics Research*, 28(1):104–126, 2009. doi: 10.1177/0278364908097884. URL https://doi.org/10.1177/0278364908097884.

[13] J. Hoffmann and B. Nebel. The ff planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, 14:253–302, May 2001. ISSN 1076-9757. doi: 10.1613/jair.855. URL http://dx.doi.org/10.1613/jair.855.

[14] Steven M LaValle. Rapidly-exploring random trees: A new tool for path planning. *TR 98-11, Computer Science Dept., Iowa State University*, 1998.

[15] Lydia E Kavraki, Petr Svestka, Jean-Claude Latombe, and Mark H Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE*

Transactions on Robotics and Automation, 12(4):566–580, 1996.

[16] Caelan Reed Garrett, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Pddlstream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning. In *International Conference on Automated Planning and Scheduling (ICAPS)*, volume 30, pages 440–448, 2020.

[17] Michael Cashmore, Maria Fox, Derek Long, and Daniele Magazzeni. Smtplan: A smt-based planner. In *International Conference on Automated Planning and Scheduling (ICAPS)*, 2015.

[18] Xiaohan Zhang, Yifeng Zhu, Yan Ding, Yuke Zhu, Peter Stone, and Shiqi Zhang. Visually grounded task and motion planning for mobile manipulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 1925–1931. IEEE, 2022.

[19] Xiaohan Zhang, Yan Ding, Yohei Hayamizu, Zainab Altaweel, Yifeng Zhu, Yuke Zhu, Peter Stone, Chris Paxton, and Shiqi Zhang. Llm-grop: Visually grounded robot task and motion planning with large language models. *The International Journal of Robotics Research*, page 02783649251378196, 2025.

[20] Léonard Jaillet, Juan Cortés, and Thierry Siméon. Transition-based rrt for path planning in continuous cost spaces. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2145–2150, 2010.

[21] Leslie Pack Kaelbling and Tomás Lozano-Pérez. Integrated task and motion planning in belief space. *The International Journal of Robotics Research*, 32(9-10):1194–1227, 2013.

[22] Caelan Reed Garrett, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Ffrob: Leveraging symbolic planning for efficient task and motion planning. *The International Journal of Robotics Research*, 37(1):104–136, 2018.

[23] Sundar Pichai, Jeff Dean, Barret Zoph, Oriol Vinyals, Chandan Singh, et al. Gemini: A family of highly capable multimodal models, 2023.

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning

transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021.

[25] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:23716–23736, 2022.

[26] OpenAI. Gpt-4v(ision) system card. Technical report, OpenAI, 2023.

[27] Sichao Liu, Jianjing Zhang, Robert X. Gao, Xi Vincent Wang, and Lihui Wang. Vision-language model-driven scene understanding and robotic object manipulation. In *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*, pages 21–26, 2024. doi: 10.1109/CASE59546.2024.10711845.

[28] Venkatesh Sripada, Samuel Carter, Frank Guerin, and Amir Ghalamzan. Scene exploration by vision-language models, 2025. URL https://arxiv.org/abs/2409.17641.

[29] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024. URL https://arxiv.org/abs/2401.14159.

[30] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[31] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023.

[32] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *International Conference on Learning Representations (ICLR)*, 2022.

[33] Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Harold Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 36067–36080, 2022.

[34] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. In *Conference on Robot Learning (CoRL)*, pages 1769–1782, 2022.

[35] Tom Silver, Varun Hariprasad, Reece S Shuttleworth, Nishanth Kumar, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Generalized planning in pddl domains with pretrained large language models. *AAAI Conference on Artificial Intelligence*, 37(12):15305–15314, 2023.

[36] Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. Llm+ p: Empowering large language models with optimal planning proficiency. *arXiv preprint arXiv:2304.11477*, 2023.

[37] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning (CoRL)*, pages 2165–2183, 2023.

[38] Yan Ding, Xiaohan Zhang, Saeid Amiri, Nieqing Cao, Hao Yang, Andy Kaminski, Chad Esselink, and Shiqi Zhang. Integrating action knowledge and llms for task planning and situation handling in open worlds. *Autonomous Robots*, 47(8):981–997, 2023.

[39] Xiaohan Zhang, Zainab Altaweel, Yohei Hayamizu, Yan Ding, Saeid Amiri, Hao Yang, Andy Kaminski, Chad Esselink, and Shiqi Zhang. Dkprompt: Domain knowledge prompting vision-language models for open-world planning. *arXiv preprint arXiv:2406.17659*, 2024.

[40] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. *arXiv preprint arXiv:2401.06209*, 2024.

[41] Tianrui Liu, Wenxuan Geng, Qinghao Zhu, Tianfei Liu, Wentao Li, Chao Ma, Qifeng Li, Juncheng Yan, and Xihui Lu. Hallusionbench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. *arXiv preprint arXiv:2310.14566*, 2023.

[42] Chaoyi Wu, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. Next-generation large vision-language models: Empirical analysis and future directions. *arXiv preprint arXiv:2306.13394*, 2023.

[43] Zi Lin, Jeremiah Garg, Tuan Vu, and Mohit Bansal. Generating calibrated uncertainty for vision-language models. *NeurIPS Workshop on Robustness of Few-shot and Zero-shot Learning in Foundation Models*, 2023.

[44] Kaizhi Zheng, Xiaotong Chen, Odest Jenkins, and Xin Eric Wang. VLMbench: A compositional benchmark for vision-and-language manipulation. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=NAYoSV3tk9.

[45] Ruzena Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):966–1005, 1988.

[46] Jeannette Bohg, Karol Hausman, Bharath Sankaran, Oliver Brock, Danica Kragic, Stefan Schaal, and Gaurav S. Sukhatme. Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics*, 33(6):1273–1291, December 2017. ISSN 1941-0468. doi: 10.1109/tro.2017.2721939. URL http://dx.doi.org/10.1109/TRO.2017.2721939.

[47] Simon Kriegel, Tim Bodenmüller, Michael Suppa, and Gerd Hirzinger. A surface-based next-best-view approach for automated 3d model completion of unknown objects. In *2011 IEEE International Conference on Robotics and Automation*, pages 4869–4874, 2011. doi: 10.1109/ICRA.2011.5979947.

[48] Jonathan Daudelin, Gangyuan Jing, Tarik Tosun, Mark Yim, Hadas Kress-Gazit, and Mark Campbell. An integrated system for perception-driven autonomy with modular robots. *Science Robotics*, 3(23):eaat4983, 2018. doi: 10.1126/scirobotics.aat4983. URL https://www.science.org/doi/abs/10.1126/scirobotics.aat4983.

[49] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements

in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(2), 2008.

[50] C. Stachniss, D. Hahnel, and W. Burgard. Exploration with active loop-closing for fastslam. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 2, pages 1505–1510 vol.2, 2004. doi: 10.1109/IROS.2004.1389609.

[51] Michael Danielczuk, Matthew Matl, Saurabh Gupta, Andrew Li, Andrew Lee, Jeffrey Mahler, and Ken Goldberg. Segmenting unknown 3d objects from real depth images using mask r-cnn trained on synthetic data. In *2019 International Conference on Robotics and Automation (ICRA)*, page 7283–7290. IEEE Press, 2019. doi: 10.1109/ICRA.2019.8793744. URL https://doi.org/10.1109/ICRA.2019.8793744.

[52] Andy Zeng, Shuran Song, Stefan Welker, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Learning synergies between pushing and grasping with self-supervised deep reinforcement learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4238–4245. IEEE, 2018.

[53] Blai Bonet and Hector Geffner. Planning with incomplete information as heuristic search in belief space. In *Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems*, pages 52–61, 2000.

[54] Jörg Hoffmann and Ronen I Brafman. Towards reasonable and forceful contingent planning. *Journal of Artificial Intelligence Research*, 22:479–527, 2005.

[55] Hanna Kurniawati and Vinay Yadav. An online pomdp solver for uncertainty planning in dynamic environment. In *Robotics Research: The 16th International Symposium ISRR*, pages 611–629. Springer, 2016.

[56] Sven Koenig and Maxim Likhachev. Fast replanning for navigation in unknown terrain. *IEEE transactions on robotics*, 21(3):354–363, 2005.

[57] Sung Wook Yoon, Alan Fern, and Robert Givan. Ff-replan: A baseline for probabilistic planning. In *International Conference on Automated Planning and Scheduling*, 2007. URL https://api.semanticscholar.org/CorpusID:15013602.

[58] Roman Van der Krogt and Mathijs De Weerdt. Plan repair as an extension of planning. In *International Conference on Automated Planning and Scheduling (ICAPS)*, pages 161–170, 2005.

[59] M. Fox, D. Long, and D. Magazzeni. Plan-based policies for efficient multiple battery load management. *Journal of Artificial Intelligence Research*, 44:335–382, June 2012. ISSN 1076-9757. doi: 10.1613/jair.3643. URL http://dx.doi.org/10.1613/jair.3643.

[60] Ronen I. Brafman, Jörg Hoffmann, and Henry A. Kautz. Continual planning and learning. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI'11, pages 1426–1431, San Francisco, California, USA, 2011. AAAI Press. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI11/paper/view/3757.

[61] Alfred A Rizzi and Daniel E Koditschek. A hybrid position/force controller for manipulation. *IEEE Transactions on Robotics and Automation*, 10(4):462–474, 1994.

[62] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In *Advances in Neural Information Processing Systems*, volume 33, pages 4247–4258, 2020.