UNIVERSITÄT BONN

Masterarbeit
zur Erlangung des Grades
Master of Science (M. Sc.)
der Landwirtschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn
Institut für Geodäsie und Geoinformation

Active Perception and Mapping for Open Vocabulary Object Goal Navigation

von

Utkarsh Bajpai

aus
Mumbai, India

Bonn 2025

Supervisor:
Prof. Dr. Cyrill Stachniss, University of Bonn, Germany

Second Supervisor:
Dr. Marija Popović, TU Delft, Netherlands
Julius Rückin, University of Bonn, Germany

Statement of Authorship

I hereby certify that this master thesis has been composed by myself. I have not made use of the work of others or presented it here unless it is otherwise acknowledged in the text. All references and verbatim extracts have been quoted, and all sources of information have been specifically acknowledged.

Place, Date	(Signature)

Acknowledgments

This journey, filled with late nights and boundless questions, would not have been possible without the light of guidance, inspiration, and unwavering support I found along the way. First, to my advisors, Dr. Marija Popović and Julius Rückin for granting me the independence to forge my own path in this project, for their steadfast guidance in seeing it through, and for their invaluable help in refining my writing. I am especially grateful to Julius for his insights, which sharpened the mathematical rigor of this work and brought clarity to complex ideas.

I extend my heartfelt gratitude to Prof. Dr. Cyrill Stachniss, not only for supporting my research on this topic but also for imparting much of the foundational knowledge I applied in this thesis. His expertise and passion for teaching laid the groundwork for my understanding and curiosity in the field. I am also grateful to Dr. Jens Behley for introducing me to deep learning and offering support throughout my studies.

A special thanks to Dr. Carlos Carbone for providing me with time during my internship to delve into the foundational ideas behind this work, and for believing in my vision. My sincere appreciation goes to Federico Magistri and Dr. Tiziano Guadagnino, who offered direction and insight during my master's project. I am also grateful to Murad Dawood, Elias Marks, and Josef Baumert, whom I assisted as a student. Their support not only enriched my learning and professional experience but also provided financial assistance that helped fund my studies. My appreciation also extends to Thomas Läbe for teaching me invaluable system administration skills and for setting up the computers that allowed my work to flow seamlessly.

Yibin, you have been an incredible mentor and flatmate, and I am deeply grateful for your support. To Fares, Arush, Pugs, and Tanmoy, thank you for your emotional support, for making music together, and for cooking meals that sustained both body and spirit. Special thanks to Pugs and Tanmoy for the countless discussions about the project and our brainstorming sessions. To Aibek, my comrade throughout the six months of this thesis, I am endlessly grateful for your companionship. Thank you, Anand and Alvin, for your support in tough times, and Prof. Muchkund Dubey for inspiring me to broaden my horizons.

Finally, to my family—Baba, Papa, and Mummy—who have supported me unwaveringly throughout my life and made it possible for me to come to Germany. And to Sylvia, without whom I might not be writing this thesis today; your support has been my greatest strength.

In each page of this work, I carry the essence of these invaluable relationships and contributions. For this, I am profoundly grateful.

Abstract

Household robots can elevate the quality of life by automating routine tasks, particularly finding user-specified items within the home. This capability is essential for robots performing tasks such as tidying up, cooking, or assisting individuals with limited mobility. This thesis addresses the challenging task of finding user-specified objects in indoor environments. In this task, a robot must locate a target object based on a natural language description provided by the user in an initially unknown environment. This task presents two main challenges. First, it requires reliable detection across a wide range of items a user may request, creating a perception challenge. Second, the robot must locate these items within cluttered, varied room layouts, adding an exploration challenge.

Recent research suggests that semantically guided, goal-directed exploration can improve the efficiency of locating a target object by guiding the robot toward target-relevant areas. State-of-the-art methods for this problem leverage vision-language models (VLMs) to provide semantic guidance by comparing observed images of regions with the linguistic description of the target object, thereby estimating semantic similarity. However, these approaches often overlook the inherent ambiguity in natural language descriptions, which introduces uncertainty into VLM-based predictions. Failing to incorporate this uncertainty can lead to overconfidence, misdirected exploration, and reduced success in locating the target object. Additionally, VLM-guided exploration approaches often employ myopic strategies, focusing on exploring the most semantically similar region at each step without explicitly accounting for future observations. While effective in many cases, this approach may still have limitations in complex environments.

We present a novel semantic uncertainty-informed active perception framework to address these challenges. Our framework integrates perception, mapping, and planning for effective object search in household environments. We leverage VLMs for perception, enabling the robot to understand and identify arbitrary objects in the environment based on natural language descriptions. Recognizing the inherent uncertainty in VLM-based perception due to linguistic ambiguities, we present a method to quantify this uncertainty by generating a range of linguistic descriptions that convey the same semantic context but capture diverse interpre-

tations. Using this uncertainty, we construct a probabilistic metric-semantic map that guides exploration based on the estimated semantic similarity of the target object to the various regions in the environment.

Our contributions are threefold. First, we propose a method to quantify the uncertainty in semantic similarity derived from VLM-based perception. Second, we develop a probabilistic map that captures uncertainty in semantic similarity. Third, to evaluate the effectiveness of our framework in finding objects, we develop both myopic and non-myopic planners that utilize this map for exploration. Including both approaches allows us to assess how each strategy performs under uncertainty, particularly in complex environments where exploration demands a balance of immediate and future-oriented decisions. Our planners employ an information-theoretic reward function to balance exploiting regions with high expectation of semantic similarity with exploration of regions with high uncertainty. Experimental evaluations demonstrate that our approach achieves comparable or marginally lower success rates than state-of-the-art approaches on this task while performing uncertainty-informed exploration. Finally, we open-source our code for usage by the community.

Zusammenfassung

Haushaltsroboter können die Lebensqualität erhöhen, indem sie Routineaufgaben automatisieren, insbesondere das Auffinden von benutzerdefinierten Gegenständen im Haushalt. Diese Fähigkeit ist wichtig für Roboter, die Aufgaben wie Aufräumen, Kochen oder die Unterstützung von Personen mit eingeschränkter Mobilität übernehmen. Diese Arbeit befasst sich mit der anspruchsvollen Aufgabe, benutzerdefinierte Objekte in Innenräumen zu finden. Bei dieser Aufgabe muss ein Roboter ein Zielobjekt auf der Grundlage einer vom Benutzer in natürlicher Sprache gegebenen Beschreibung in einer zunächst unbekannten Umgebung finden. Diese Aufgabe stellt zwei große Herausforderungen dar. Erstens erfordert sie eine zuverlässige Erkennung einer Vielzahl von Objekten, die ein Benutzer anfordern kann, was eine Herausforderung für die Wahrnehmung darstellt. Zweitens muss der Roboter diese Gegenstände in einem unübersichtlichen, vielfältigen Raumlayout finden, was eine zusätzliche Herausforderung für die Erkundung darstellt.

Jüngste Forschungsergebnisse deuten darauf hin, dass eine semantisch geführte, zielgerichtete Erkundung die Effizienz der Lokalisierung eines Zielobjekts verbessern kann, indem der Roboter auf zielrelevante Bereiche gelenkt wird. Modernste Methoden für dieses Problem nutzen Vision-Language-Modelle (VLMs), um semantische Führung zu bieten, indem sie beobachtete Bilder von Regionen mit der sprachlichen Beschreibung des Zielobjekts vergleichen und so die semantische Ähnlichkeit schätzen. Bei diesen Ansätzen wird jedoch häufig die Mehrdeutigkeit von Beschreibungen in natürlicher Sprache übersehen, was zu Unsicherheiten bei VLM-basierten Vorhersagen führt. Wird diese Unsicherheit nicht berücksichtigt, kann dies zu übermäßigem Vertrauen, fehlgeleiteter Erkundung und geringerem Erfolg beim Auffinden des Zielobjekts führen. Darüber hinaus verwenden VLM-basierte Erkundungsansätze oft kurzsichtige Strategien, die sich bei jedem Schritt auf die Erkundung der semantisch ähnlichsten Region konzentrieren, ohne zukünftige Beobachtungen explizit zu berücksichtigen. Dieser Ansatz ist in vielen Fällen effektiv, kann jedoch in komplexen Umgebungen dennoch limitierend sein.

Wir stellen ein neuartiges, auf semantischer Unsicherheit basierendes aktives Wahrnehmungssystem vor, um diese Herausforderungen zu bewältigen. Unser System integriert Wahrnehmung, Kartierung und Planung für eine effektive Ob-

jektsuche in häuslichen Umgebungen. Wir nutzen VLMs für die Wahrnehmung, die es dem Roboter ermöglichen, beliebige Objekte in der Umgebung auf der Grundlage von Beschreibungen in natürlicher Sprache zu verstehen und zu identifizieren. Wir erkennen die inhärente Unsicherheit in der VLM-basierten Wahrnehmung aufgrund von sprachlichen Mehrdeutigkeiten und stellen eine Methode zur Quantifizierung dieser Unsicherheit vor, indem wir eine Reihe von sprachlichen Beschreibungen erzeugen, die denselben semantischen Kontext vermitteln, aber unterschiedliche Interpretationen erfassen. Unter Verwendung dieser Unsicherheit konstruieren wir eine probabilistische metrisch-semantische Karte, die die Erkundung auf der Grundlage der geschätzten semantischen Ähnlichkeit des Zielobjekts mit den verschiedenen Regionen in der Umgebung leitet.

Unsere Beiträge sind vielfältig. Erstens schlagen wir eine Methode zur Quantifizierung der Unsicherheit in der semantischen Ähnlichkeit vor, die sich aus der VLM-basierten Wahrnehmung ergibt. Zweitens entwickeln wir eine probabilistische Karte, die die Unsicherheit in der semantischen Ähnlichkeit erfasst. Drittens entwickeln wir zur Bewertung der Effektivität unserer Methode beim Auffinden von Objekten sowohl kurzsichtige als auch nicht kurzsichtige Wegplanungsmethoden, die diese Karte zur Erkundung verwenden. Die Einbeziehung beider Ansätze ermöglicht es uns, zu beurteilen, wie jede Strategie unter Unsicherheit abschneidet, insbesondere in komplexen Umgebungen, in denen die Erkundung ein Gleichgewicht zwischen unmittelbaren und zukunftsorientierten Entscheidungen erfordert. Unsere Wegplanungsmethoden verwenden eine informationstheoretische Belohnungsfunktion, um ein Gleichgewicht zwischen der Erkundung von Regionen mit hoher erwarteter semantischer Ähnlichkeit und der Erkundung von Regionen mit hoher Unsicherheit herzustellen. Experimentelle Auswertungen zeigen, dass unser Ansatz bei dieser Aufgabe vergleichbare oder geringfügig niedrigere Erfolgsquoten als State-of-the-Art-Ansätze erzielt, während eine auf Unsicherheit basierende Erkundung durchführt wird. Abschließend stellen wir unsere Implementierung der Methode der Community als Open Source zur Verfügung.

Task Description

Building robots that operate in households as personal assistants is a longstanding goal of the field of robotics and artificial intelligence. Embodied AI has recently emerged as a research field that emphasizes the usage of AI techniques, such as computer vision and natural language processing within physical entities, to achieve this goal. The survey by Srivastava et al.[109] reports that the top 100 tasks humans want robots to perform in their houses revolve around cleaning, cooking, and rearranging objects. A prerequisite to achieving such composite tasks is the capability to navigate to specified objects present in the environment autonomously. This is referred to as the object goal navigation (ObjectNav) task in literature [6] and serves as the focal point of this thesis. Large variations across different households and complexity in terms of their layouts, structures, and objects necessitate a robot to understand the geometric and semantic aspects of its environment for planning. Previous works on indoor robot scene understanding have focused on using particle filter-based mapping and localization approaches [36] and metric-semantic maps [45, 7, 37] to map and localize in an environment accurately. For indoor navigation, sampling-based path planners [50], deep reinforcement learning [118], and active sensing using informative path planning [115] have been used. Most approaches rely on deep learning-based perception [92, 89]. Recently, works like Gadre et al. [30, 52] proposed utilizing large-scale transformer-based foundational models for perception in scene exploration, specifically open vocabulary feature detection [85, 79] and segmentation algorithms [54], that provide detection and segmentation of objects with natural language labels in an open world setting unlike previous deep learning methods, which has led to their utilization as a perception backbone for ObjectNav.

Approaches to tackle the open vocabulary ObjectNav task usually consider the environment to be map-based or map-less. In the map-based scenario, works focus on map construction while navigation is considered a downstream task. Liu et al. [68] demonstrate object search and navigation on an open vocabulary 3D reconstructed static map. Hughes et al. [45] introduce a real-time closed-vocabulary 3D scene graph representation for mapping. Gu et al. [37] improve this by creating an open vocabulary metric-semantic 3D scene graph, but their approach is not real-time. Another notable state-of-the-art approach is VL-Maps

by Huang et al. [44] where the authors create a 2D metric semantic map by fusing features from an open vocabulary segmentation network with RGBD camera observations, which can also be used for obstacle avoidance. However, planning is only possible after the environment is mapped and scene exploration is not dealt with in this approach. On the other hand, in the map-less scenario, research focuses on exploration with active perception to locate the object. Recent works in this area employ vision language models (VLMs) as object detectors [52, 30, 103], develop transformer-based architectures [122, 27, 10], and use LLMs for planning [103]. Yokohama et al. [128] develop VLFM to semantically bias frontier-based exploration based on the inductive biases of VLMs, allowing them to explore regions most correlated to the desired object. However, a primary drawback of their approach is the use of handcrafted update functions instead of probabilistic map updates for likelihood estimation. Additionally, their reliance on frontier-based exploration, a myopic planning strategy, optimizes only for the immediate action without considering future observations.

This project will aim to develop an approach for addressing the ObjectNav problem in household environments. The approach will search for a static object by actively exploring a static 3D environment having no prior map. During exploration, it will build a probabilistic open vocabulary map which will be used for active re-planning. As starting points, exploration and map building will be performed similarly to VLFM. To be consistent with the assumptions of the above method, RGB-D images from the forward-facing camera of a mobile robot and its error-free poses are assumed as inputs. The project will address the limitations of baseline methods by developing a hybrid exploration and mapping pipeline which will aim to find arbitrary objects, in unfamiliar household environments. The approach will develop a probabilistic semantic map and actively explore an environment, using non-myopic strategies.

Experiments will aim to show the effects of building a probabilistic open vocabulary map and planning in a non-myopic manner during exploration on the ObjectNav task. Using this strategy, the robot will actively explore the environment to find the target object. We plan to evaluate our approach against the baseline VLFM [128] representing a state-of-the-art method for open vocabulary mapping and exploration. We will be using metrics like the success rate of finding the object, path efficiency and distance traveled [1], which will highlight the robustness and efficiency of using a hybrid mapping-exploration approach. Experiments on household scenes in a high-fidelity simulator like Habitat [82] and using real-world datasets such as the HM3D and the MP3D Dataset [86, 16] will aim to show that the method can be applied on realistic household environments and can provide a basis for ObjectNav in households.

for my grandfather

Contents

Chapter 1

Introduction

In a rapidly evolving world, robots are poised to play a pivotal role in improving quality of life by automating time-consuming household tasks, increasing convenience, and supporting independent living. Recent advancements have led to the emergence of versatile, general-purpose robots capable of performing a wide range of household tasks, including cleaning, tidying up, cooking and assisting individuals. These robots represent a shift from single-task robots, such as those designed for automatic vacuuming, to more autonomous systems, as illustrated in Figure 1.1. However, enabling robots to autonomously perform diverse household tasks presents significant challenges, which motivate the research presented in this thesis. Most household tasks, such as tidying up or preparing meals, require robots to locate and identify specific objects within cluttered environments. The diversity of household environments, with numerous types of objects and varied room layouts, makes these tasks challenging.

Addressing this challenge requires the development of methods capable of reliably locating user-specified objects in complex, unstructured environments with high efficiency. In robotics, this is formalized as the object goal navigation (ObjectNav) problem [6, 1, 112], where the objective is to direct a robot to find a specified object based on user instructions within an unfamiliar environment. This problem entails two primary difficulties. First, recognizing a potentially limitless range of objects, that exhibit diverse visual characteristics, including varying shapes and sizes. Second, efficiently searching various household areas despite the substantial uncertainty about the object's location. Given the inherent partial observability of the environment, where the robot can only perceive limited regions at a time, it must continuously make informed estimates about the target's potential location based on the information collected during the search.

Overcoming these challenges necessitates an integrated approach that combines perception and action in a mutually informative process. Perception involves two critical components: spatial understanding, which entails interpreting

Figure 1.1: Household robots are evolving from task-specific machines, such as vacuum cleaners or lawn mowers (left), to general-purpose assistants capable of tackling a diverse array of tasks such as tidying up (right). Courtesy: (from left) *iRobot, Scythe Robotics, Alphabet, Boston Dynamics, Hello Robot, 1X Technologies*

the environment's layout and geometric structure, and semantic understanding, which involves recognizing objects and leveraging contextual information to infer probable locations. These components collectively enable the robot to refine its predictions about the target's likely position as new data is obtained. However, perception alone is insufficient to resolve the inherent uncertainty. The robot must also act on this information to actively reduce ambiguity in the search process. This requires goal-directed exploration strategies that exploit both spatial and semantic insights, continuously updating the search approach as new observations are integrated. By dynamically adapting its strategy based on real-time data, this integrated perception-action framework allows the robot to navigate the variability and complexity characteristic of household environments.

In this thesis, we address the challenges of object search in household environments by proposing a framework that enables robots to locate target objects. Our approach equips robots with the ability to perceive a wide range of objects while continuously updating their understanding of the environment through a metric-semantic map, which integrates both spatial and semantic information. This map allows the robot to represent the environment in a way that accounts for perception uncertainties and guides its exploration process. By employing planning algorithms that utilize the information in the map, the system directs the robot toward regions where the target object is more likely to be found, based on estimates of uncertainty from perception. This uncertainty-aware strategy prioritizes exploration in areas with a higher probability of success, allowing the robot to adapt to the complexity and variability of household settings. The methods presented in this thesis leverage the uncertainty in the perception system not as a limitation but as an opportunity to interpret contextual clues, guiding the robot to locate objects in diverse and unpredictable environments. Our

Figure 1.2: Misplacing objects is a frustrating common occurrence in everyday life. Household robots can become effective companions to humans by helping them find misplaced objects.

approach achieves performance comparable to or minimally worse than state-of-the-art methods in object search tasks while being more theoretically principled.

1.1 Motivation

Developing intelligent robotic assistants that can find objects effectively in real-world environments remains a significant challenge in robotics [6, 49, 112]. The relevance of the task becomes evident when considering common human experiences, such as misplacing everyday objects. In everyday scenarios, people often forget where they placed essential items, such as wallets, phones, or glasses, as depicted in Figure 1.2. This can disrupt routines and cause considerable frustration. Furthermore, scenarios, such as assisting elderly individuals at home or cooking meals for hospital patients, involve composite tasks that require completing several sub-tasks. For example, cooking a meal may involve fetching various ingredients, which first requires the ability to locate them, relying heavily on a robot's object-finding skills. Therefore, a fundamental capability needed for general-purpose robots is to locate arbitrary objects in unexplored household environments.

Previous research on this problem has often used exploration strategies designed to maximize the area covered by a robot, employing metric maps to track explored regions. These methods rely on simple heuristics, such as navigating to frontiers, which are boundaries between explored and unexplored areas, or targeting map corners [126, 125, 71]. However, these approaches are inherently myopic, focusing only on immediate navigation actions without planning for future steps or considering the overall map and environment. Furthermore, they lack semantic understanding and a goal-directed approach, i.e, they do not actively incorporate information about the target object during exploration or prioritize areas that

are semantically relevant to the object. As a result, these methods often lead to inefficient exploration and excessive searching when attempting to locate specific objects. In order to explore in a goal-directed way, deep learning-based techniques [105, 127, 73, 124, 122] offer an alternative by training neural networks in simulations to learn scene representations and exploration strategies. While effective for objects encountered during training, they struggle to generalize to previously unseen objects due to reliance on a fixed set of categories used during training. This limitation is compounded by the diversity of user requests, which are specified in natural language and vary in detail, such as "find my red sweater" or "find my red sweater with white stripes". Even if the model is trained on the object "sweater", it would not be able to differentiate between these specific variations. Addressing such varied requests requires systems that can handle a wide range of object types.

Pre-trained vision-language models (VLMs) [85, 63, 62] offer a promising solution for bridging the gap between rigid training and open-ended real-world tasks. Trained on large datasets of paired visual and textual data, these models can interpret textual descriptions and associate them with corresponding visual features, enabling "open vocabulary" perception. This capability means that the number of object categories is not predetermined but can be inferred from the relationships between images and text, allowing the models to adapt to a broader range of objects. Integrating VLMs into exploration strategies provides a more flexible and semantically guided approach to finding arbitrary objects. Recent efforts have combined VLMs with traditional techniques like frontier-based exploration (FBE) [30, 52, 128, 134], where exploration focuses on frontiers predicted by the VLM to have a high likelihood of containing the target object. However, these approaches inherit the myopic limitations of FBE, which performs one step utility maximization on frontiers, without considering future actions across the entire map. In addition, these approaches employ the greedy strategy of selecting the frontier the VLM predicts most likely to lead to the target object. This assumes the model's predictions are entirely reliable, ignoring the inherent uncertainty in perception models. Since perception naturally involves some degree of uncertainty, failing to account for it can result in inefficient exploration and missed opportunities for object detection. This becomes especially relevant for multimodal perception models such as VLMs, where the sources of uncertainty stem from the individual uncertainties of the modalities- visual and linguistic. When these uncertainties converge, they are compounded, ultimately amplifying the overall uncertainty in the system. Therefore, there is a need for strategies that incorporate uncertainty awareness, enabling exploration to adapt based on the confidence in the model's predictions and leading to informative exploration.

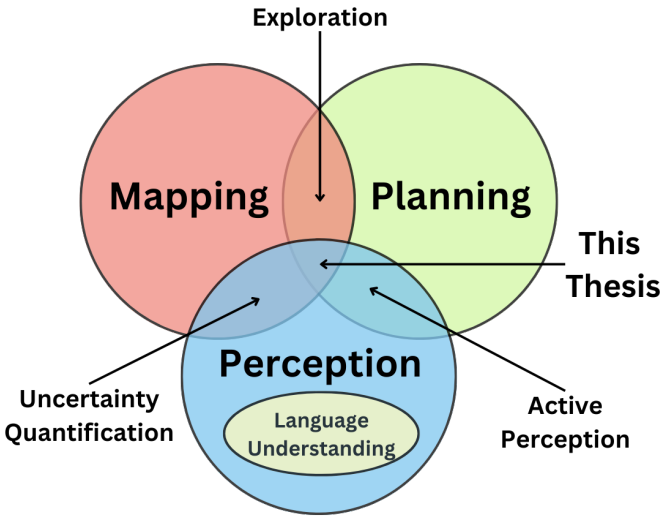


Figure 1.3: In this thesis, Perception, Mapping, and Planning form the foundation of the robot's ability to perform the Object Goal Navigation (ObjectNav) task.

1.2 Goal and Main Contributions

In this thesis, we address the ObjectNav problem by presenting a novel approach that enables robots to locate a wide range of objects in unexplored household environments based on the target object's textual description (e.g., "clothes" or "coffee table"). The main contribution of this work is a semantic uncertainty-informed active perception strategy for object search, integrating perception, mapping, and planning to locate arbitrary objects in household environments. For perception, our approach utilizes a deep learning model trained on visual and textual data to process the robot's visual feed and the object description. This allows the system to recognize a diverse range of user-specified target objects commonly encountered in household environments. However, perception is inherently uncertain due to ambiguities in language interpretation, environmental factors like lighting and viewpoint, and how the model encodes information. To manage these uncertainties, we incorporate semantic uncertainty from VLMs into the perception process, allowing the robot to identify objects based on semantic similarity while accounting for ambiguity. To support this uncertainty-aware perception, we develop a metric-semantic mapping approach that integrates both spatial and semantic information. The mapping process captures the spatial layout of the environment while incorporating the uncertainty in the semantic similarity of different regions of the environment in context of the target object. This allows the robot to represent the environment in a way that reflects these uncertainties, providing a more robust foundation for decision-making during exploration.

Building on this map representation, we design exploratory planners that guide the robot's search strategies. These planners leverage the quantified un-

certainty to focus exploration on regions with a higher likelihood of containing the target object, while accounting for the reliability of semantic predictions. By dynamically adjusting the search strategy based on the confidence in the predictions, the robot can prioritize areas with a higher probability of success, resulting in more adaptive and efficient search processes. As detailed in the following chapters, this approach equips robots to locate arbitrary objects in diverse household environments by performing uncertainty-informed active perception for object search. An overview of our approach is presented in Figure 1.3. In summary:

- We develop a training-free open vocabulary semantic uncertainty-informed active perception-based pipeline which explores an environment to search for a target object.
- We use VLMs to semantically guide exploration towards semantically similar regions, detect objects and show that they are susceptible to data uncertainty when used in downstream tasks in robotics.
- We develop an uncertainty quantification method for VLMs and create a probabilistic map representation which is more theoretically principled than current approaches.
- We develop an information theoretic reward function for two planning methodologies (myopic and non-myopic) using our probabilistic map to make uncertainty informed decisions and demonstrate that the myopic strategy performs marginally worse than state-of-the-art myopic planners which do not use a probabilistic map.

1.3 Overview of the Thesis

This thesis aims to develop a training-free, open vocabulary semantic uncertainty-informed active perception approach for addressing the ObjectNav task. A broad overview of related work in this area is provided in Chapter 2. Chapter 3 presents the fundamental techniques in perception, mapping, and planning that form the foundation of this thesis. A detailed explanation of our proposed approach is outlined in Chapter 4. Chapter 5 offers a comprehensive evaluation, including both quantitative and qualitative analyses, comparing our approach to state-of-the-art ObjectNav methods and discussing its limitations and failure modes. Additionally, we plan to open-source the complete pipeline to support and advance research on ObjectNav within the robotics community.

Chapter 2

Related Work

Active perception—a theory rooted in cognitive science—posits that agents actively engage with their environment, using prior knowledge to enhance perception. This idea has become increasingly popular in the field of robotics in recent years, allowing robots to exploit prior knowledge to achieve goal-directed behavior. In this chapter, we first present a brief introduction to active perception and relate to how it is useful in solving the object goal navigation problem. We then explore literature relevant to the core problem of this thesis: open vocabulary ObjectNav, a challenging task that integrates multiple robotics sub-fields such as perception, mapping, and planning.

The origins of active perception can be traced back to Helmholtz's foundational theory of perception [42] which proposed that perception is an inferential process driven by both top-down (sensory-driven) and bottom-up (knowledge-driven) counter-streams of processing. This theory was further developed in psychology and computational neuroscience [28, 80], where researchers highlight that perception and action are interlinked. In addition to passively perceiving the environment, an agent actively seeks relevant or goal-directed information to enhance its internal world model, in turn enhancing its perception ability. In robotics, one of the earliest references to active perception is attributed to Ruzena Bajcsy's pioneering work in 1988 on gaze control [2]. Her research introduced active perception to the field by combining a bottom-up, data-driven approach for image recognition with a top-down, knowledge-driven method for data acquisition within a single pipeline, subsequently extending it to robot path planning and exploration [4, 3]. Despite the potential, Bajcsy's work did not see widespread adoption initially due to the limitations of image processing technology at the time, which struggled to deal with high-level concepts like object recognition or semantic understanding. Instead, research continued to focus primarily on bottom-up perception, extracting low-level information from sensory data, e.g., identifying edges, corners, or obstacles[106, 40, 95, 70], for downstream tasks.

In the last decade, the advent of deep learning has revitalized interest in active perception within the robotics domain. Deep learning techniques, particularly convolutional neural networks (CNNs) [57], are trained on large-scale labeled datasets and can learn complex visual representations directly from data. This enables them to learn priors corresponding to high-level concepts such as objects. CNNs have demonstrated exceptional performance in tasks like image classification, object detection, and semantic segmentation, outperforming traditional image processing methods. Object detection, for instance, involves identifying both the position and category of objects within an image. This is commonly achieved using CNN models trained on image datasets, with annotated classes and bounding boxes. An extensive dataset which is widely used for training object detectors is the MS-COCO dataset [66], which has 80 object classes and 1.5 million object instances. Two-stage approaches to object detection such as region-based CNNs (RCNNs) [34, 33, 91], first extract regions that likely contain objects and then classify these regions, providing a bounding box, object class, and class probability. While these methods are accurate and robust, they tend to be computationally expensive and slow for inference. To address this, Redmon introduced the YOLO architecture [87] which utilizes a single-stage approach. By dividing each image into grids and detecting objects per grid cell, YOLO achieves faster inference at some cost to accuracy. Various improvements of this architecture have since enhanced both accuracy and efficiency [88, 31, 61].

However, when considering the challenges of deploying these models in real-world robotics tasks such as ObjectNav, in which a robot must navigate to any object specified by the user, the limitation to a fixed set of object classes becomes a critical constraint. In real household environments, the number of object categories far exceeds the 80 classes available in MS-COCO. This limitation stems from the supervised training paradigm, which restricts the model's capability to classify only those object classes that it is explicitly trained on. In addition, in order to generate high-quality data for supervised learning, labor-intensive and costly human annotated data is required. Consequently, while these models perform well on in-distribution classes, they struggle with out-of-distribution classes, significantly reducing their effectiveness in more diverse real-world scenarios. To overcome the limitations of traditional supervised methods for object detection, open-set perception was developed, which treats new classes as background classes during training and classifies them under a common class during inference. To generate different class labels for each newly encountered object, zero-shot learning (ZSL) driven object detection was proposed by Bansal et al. [5] which extends a detector to generalize from in-distribution classes to out-of-distribution classes,using multi-modal learning. The approach projects both images and class labels embeddings into a common vector space, exploiting semantic relationships

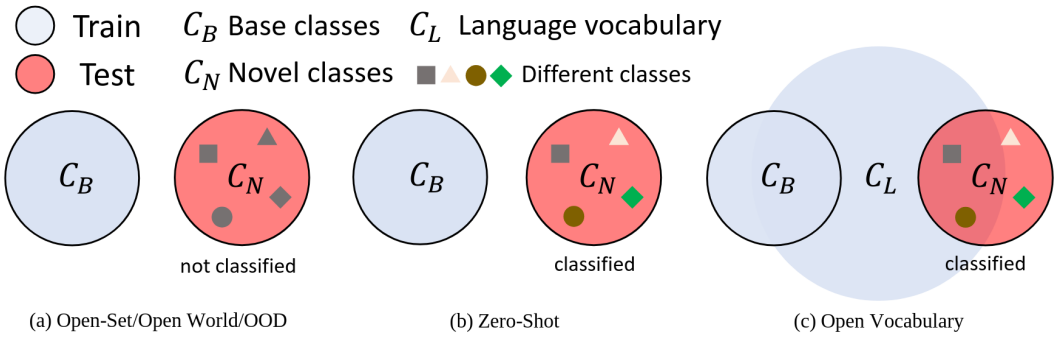


Figure 2.1: Concept of Open Vocabulary Detection. Courtesy: *Wu et al.* [119]

between in-distribution and out-of-distribution class labels to transfer a model trained on in-distribution classes to out-of-distribution classes. However, these methods lack examples of unseen objects and treat these objects as background objects during training. As a result, during inference, the model identifies novel classes solely based on their pre-defined text embeddings, limiting exploration of the visual information and relationships of those unseen classes. Open vocabulary learning, particularly in object detection and segmentation trains on a subset of base classes and allows for inference on both base and novel classes. A key advancement over zero-shot learning is the incorporation of visually related language data, as auxiliary supervision. This use of language data requires less labeling effort, making it a cost-effective alternative to traditional annotations, while introducing a much broader and flexible vocabulary, which enables models to generalize to novel classes. An analysis of the three learning paradigms is shown graphically in Figure 2.1

With the advent of the transformer architecture, cross-modal learning, which integrates data from different modalities such as text and images, has become the dominant method for a variety of core perception tasks including segmentation and object detection. Advances in cross-modal training have also driven the development of image-text matching models, which align images with their corresponding textual descriptions. A typical image-text matching pipeline consists of a visual semantic encoder, as described by Frome et al. [29]. To enhance generalization, unsupervised or weakly supervised methods, like Vision-Language Models like Contrastive Language–Image Pre-training (CLIP) [85] were introduced. CLIP was trained on a large dataset of carefully curated 400 million image-text pairs, allowing it to generalize beyond the closed-set classification problem, performing well on unseen objects and environments. For instance, CLIP claims Resnet50-level accuracy on the ImageNet dataset [97], without being trained on any of the images from the dataset. This makes it very relevant for robotics applications, where a robot might encounter a lot of different objects which fall

outside of the class of any object dataset like MS-COCO. Other approaches like BLIP [63] and BLIP-2 [62] train on frozen image and text encoders and ALIGN [47] trains on a larger noisier dataset, improving their performance on unseen objects and scenarios, demonstrating the potential of these approaches to overcome the limitations of traditional supervised method. In parallel, object detection has also witnessed innovations aimed at overcoming the limitations of traditional supervised learning methods.

Even though these models have generalization to out-of-distribution classes, on the MS-COCO dataset, they underperform models like YOLOv7 [116] that are explicitly trained on it. In our approach for ObjectNav, detecting the target object is a necessary criterion for the successful termination of the object search process. We have a hybrid object detection system, which switches between GroundingDINO for open vocabulary object detection i.e, for classes outside of the MS-COCO classes and YOLO v7, which is used to detect objects that pertain to the MS-COCO classes due to its superior accuracy on the dataset. By incorporating state-of-the-art object detection methods for both in-distribution and out-of-distribution detection, we ensure a robust and comprehensive solution to object detection for ObjectNav.

The ObjectNav task, aimed at searching for objects in an unexplored environment, is intrinsically linked to active perception as it involves goal-directed navigation based on prior beliefs, which are possible to learn with deep learning methods. ObjectNav approaches generally follow two paths: end-to-end learning or modular approaches. End-to-end learning [124, 14, 127, 73] directly maps inputs to actions and learns both visual representations and navigation policies simultaneously. Such approaches are usually trained in simulation environments due to the cheap data generation capabilities of realistic simulators like Habitat and Thor [113, 56]. In such frameworks, the robot encodes observations to extract features and embeds the goal and previous action for the policy network input. The policy network then learns the action decision-making based on the reward collected during training. These methods often integrate various visual representations, such as RGB images, bounding boxes, and semantic segmentation masks, into the policy network [67]. Shen et al. [105] utilize different representations as inputs to different policies and then fuse actions at the end. Yadav et al. [124, 122] learn visual representations offline from images of indoor environments using self-supervised learning, and then fine-tune for ObjectNav. A line of works [14, 24, 25] explore scene-object relationships, like room layout and object occurrences to localize the target object based on its similarity with them. However, end-to-end methods face challenges such as sample inefficiency, complex training regimes, and poor generalization to new environments. In addition, they struggle to generalize to out-of-distribution objects.

Modular approaches to ObjectNav [30, 128, 134, 130, 19] are different from end-to-end architectures as they do not have a single architecture that maps observations to actions. Instead they have multiple components like perception, mapping, long horizon planning and point-to-point navigation which together make the whole pipeline work. Modular methods often feature a mapping module that constructs a representation of the environment, a policy module that decides on long-term goals, and a local path-planning module that outputs specific actions, where some components are learned while others rely on classical methods. Scene representations in these systems can take various forms, ranging from feature-based maps that store semantic features to metric representations like grid maps [77, 110]. Some approaches also represent the environment in terms of hierarchies of smaller sub maps or graphs which helps reduce the computational burden of mapping and planning in extensive environments [43, 7, 93, 94, 45]. The integration of vision-language models has further enhanced the richness of these representations, allowing for maps that could encode semantic information based on open vocabulary features. A leading example is VL-Maps by Huang et al. [44] which constructs an 2D metric-semantic top-down map by merging features from an open vocabulary segmentation network with geometric data. Other approaches [18, 68] develop similar open vocabulary maps. Moreover, several methods build scene graphs using open vocabulary features [37, 117]. However, such dense maps are usually computationally expensive and are not built incrementally in real time, so they cannot be used for online exploration but can be used to navigate to objects after the map has been created.

In order to search for an object, a global planner or policy is needed. Long horizon planning policies in modular methods typically aim to explore the initially unknown environment. In their work SemExp, Chaplot et al. [17] introduce a framework using a 2D semantic segmentation network to build explicit local gridmaps, which are accumulated to get a global top down semantic grid map of the environment. Their learned navigation policy uses both local and global maps to predict a long term goal for navigation. Other works like Stubborn [71] choose a simpler exploration objective, by navigating from corner to corner in the map. Rudra et al. [96] sample viewpoints from a 2D occupancy grid map and compute the probability of spotting the target at these points. A large number of works utilize the frontier-based exploration (FBE) policy proposed by Yamauchi [126, 125, 13]. FBE operates on the principle that moving towards the boundary between known and unknown areas yields the most new information about the environment. In FBE, a robot incrementally expands its known environment by navigating to the boundary (or frontier) between known and unknown regions. Zero Shot ObjectNav methods like Clip on Wheels (CoW) by Gadre et al. [30] follow the FBE approach [126] and navigate to the closest unvisited frontier from

the robot's position until the target object is detected using CLIP features or an open vocabulary object detector. However, uninformed greedy exploration biases the policy to explore low importance regions. To counteract this, many methods for choosing the next frontier have been proposed, such as classical methods that select frontiers based on the expected amount of information a robot would gain, based on the number of frontier cells and their distance from the robot [75]. Li et al. [65] develop a method to estimate the expected reward at frontiers to informatively decide which frontier to navigate to. They train a CNN to estimate properties associated with each frontier like the expected unexplored area beyond each frontier and expected time steps required to explore it and focus on enhancing coverage of indoor environments. Dorbala et al. [22], Yu et al. [130] and Zhou et al. [134] use large language models like GPT-3 [11] to infer common sense relations between objects and rooms to select frontiers that would be valuable to find the target object. Chen et al. present SemUtil [19] and use BERT to embed class labels of objects detected near the frontiers and then compare them to the text embedding of the target object to select the frontier to explore next. Yokohama et al. develop VLFM [128] which uses Vision Language Models to project cosine similarities of egocentric RGB frames to build a 2D "value" map , while simultaneously building an occupancy grid map for obstacle avoidance. They show state-of-the-art performance on the ObjectNav task and also demonstrate their approach in the real world. Since such algorithms do not balance between exploration and exploitation, an improvement was developed to incorporate Monte-Carlo tree search (MCTS) [35]. There have also been attempts to combine MCTS and frontier-based planning to help the tree search when it gets stuck in local minima [59].

Finally, to navigate from the robot's position to the navigation goal generated by the long horizon planning policy, modular approaches adopt point-goal navigation algorithms. Point-goal navigation assumes that the robot's initial state and the goal explicitly defined in the coordinate frame of the environment to plan collision free trajectories. Classical search-based planners like Dijkstra[21] or A* [41, 8] treat the initial and goal states as nodes on a tree and then use graph traversal algorithms to find the shortest path. Sampling-based tree traversal algorithms, such as Probabilistic Roadmaps [51] and Rapidly Exploring Random Tree (RRT) [60], sample points in the search space, making planning in higher dimensional and continuous action spaces tractable, alleviating the exponential computational complexity. Variants of these approaches [50, 46] claim faster convergence while guaranteeing asymptotic optimality. Deep learning-based end-to-end approaches such as reinforcement learning or imitation learning learn to directly map observations to actions and have been used for waypoint planning [137, 129, 58, 74, 102]. Notably, the DDPPO policy proposed by Wi-

jmans et al. [118] enables point-goal navigation using image frames and ground truth poses, eliminating the need for a map to plan, achieving a near perfect success rate and setting a new standard for indoor navigation, which has been used in many ObjectNav policies [32, 128].

In this thesis, we aim to build a modular pipeline for Zero Shot Open Vocabulary Object Goal Navigation, to develop a training-free approach to solve this problem, utilizing frozen vision-language models as priors to build a semantic map similar to VLFM [128]. We focus on uncertainty quantification of these models to enhance the map and utilize classical exploration pipelines for long horizon planning to avoid training complexities and the lack of generalization to out-of-distribution scenarios of end-to-end approaches. A concurrent work in this direction is from Ren et al. [90] which quantifies uncertainty in VLMs from a single viewpoint and uses FBE. In contrast to this, we quantify uncertainty for both viewpoint as well as text prompts in a single Bayesian framework and use an information-theoretic reward function to drive exploration.

Chapter 3

Basic Techniques

The ObjectNav task requires a robot to explore an unfamiliar environment in order to search for a user-specified target object by the user through a natural language description. Addressing this task necessitates that the robot perceives arbitrary objects in the environment, maintains a representation of its surroundings, and plans exploration based on this representation to navigate toward the target object. This chapter outlines the core techniques that form the foundation of our approach to address the challenges of this task. We begin by introducing the object goal navigation problem in Section 3.1. Given the modular nature of our active perception approach, consisting of perception, mapping, and planning components, we provide a conceptual background on these key aspects. In Section 3.2 we introduce the reader to open vocabulary perception, which we utilize to perceive arbitrary objects. We discuss grid mapping in Section 3.3 which relate to our metric semantic map representation. The chapter concludes with an overview of MCTS planning in Section 3.4, which forms the basis of our non-myopic planner.

3.1 Object Goal Navigation

The task of Object Goal Navigation or "ObjectNav" as defined by Batra et al. [6] requires an robot to navigate to an instance of a specified object category from a predefined set of object categories in an unseen environment. The robot does not receive a map of the environment and must navigate using its onboard sensors: an RGB-D camera and a GPS+Compass sensor which provides position and yaw. These sensors are assumed to be noiseless. In practice, the set of object categories are restricted to a fixed set of coarse descriptions of objects (e.g.,"cup") pertaining to the categories of the MS-COCO dataset [66]. Recent works [19, 30, 128, 22] extend this definition to allow for arbitrary natural language descriptions of objects (e.g., "cat shaped mug", "cup under the table"), which extends the set

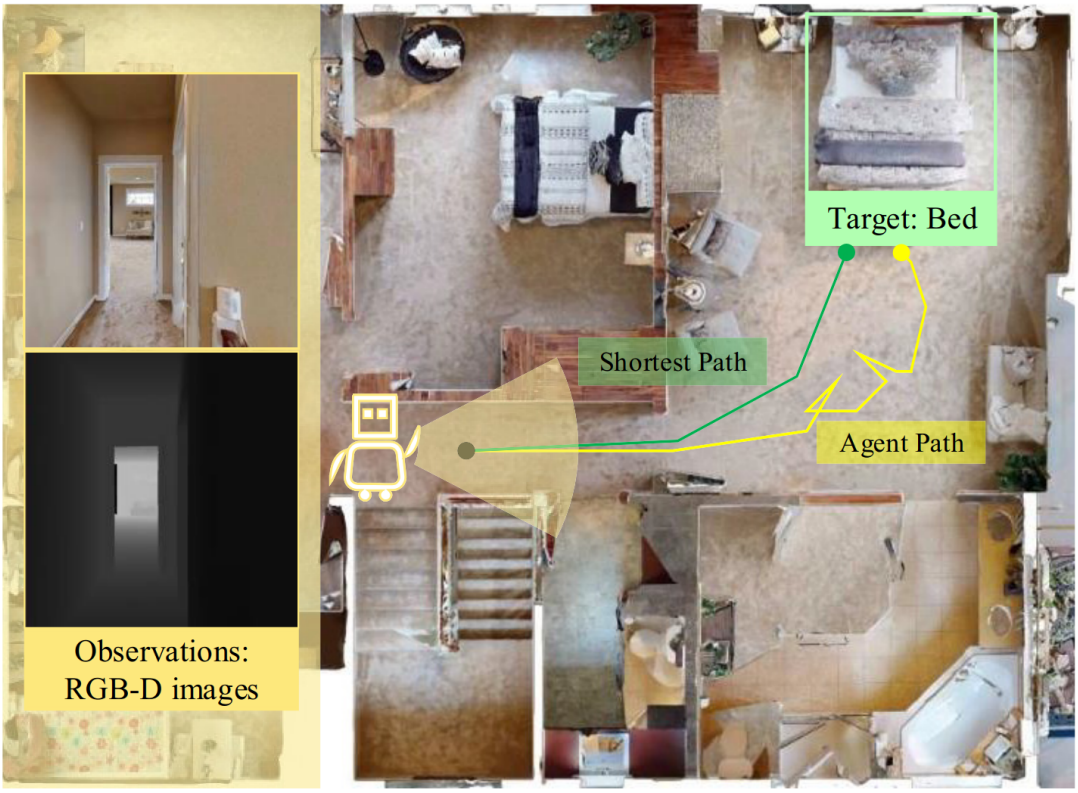


Figure 3.1: This figure shows a typical object goal navigation (ObjectNav) task in which a robot needs to explore an unknown indoor environment in order to search for an object. At each timestep the robot receives an observation from its RGB-D and GPS+Compass sensors, which it uses to perceive the environment and to localize itself. In this example, the target object that the robot needs to find is a "bed". Courtesy: *Sun et al.* [112]

of object categories to an infinite set.

We adopt this extended definition and formally define the ObjectNav task as an open-world goal-directed navigation task, where a robot is initialized in an unknown environment $e \in E$, where E denotes the set of environments. The robot is initialized at a random pose $\mathbf{x_0^w} = (v_0^w, r_0^w)$, where $v_0^w \in \mathbb{R}^3$ and $r_0^w \in SO(2)$ are the robot's starting position and rotation in the world coordinate frame w. We define the set of all object descriptions as $\mathcal{O}$ and the set of all objects in the environment as $\mathcal{G}$. Based on the natural language description of the target object $o \in \mathcal{O}$, the robot is required to navigate to an object $g \in \mathcal{G}$ which is present in the environment with pose $\mathbf{x_g^w}$ and matches the description o. There is a many-to-many relationship between the elements of $\mathcal{O}$ and $\mathcal{G}$, formally captured as $R \subseteq \mathcal{O} \times \mathcal{G}$. For instance, target object descriptions in $\mathcal{O}$ such as "chair", "brown chair", "dining chair" can all describe the object "chair" in $\mathcal{G}$ if it has matching characteristics. Additionally, there can be many instances of g present in the environment, which fit the description of o, i.e., there can be many chairs in the environment that match the target object "chair".

The goal of ObjectNav is to navigate to g as described by o. In order to find g, the robot explores the environment, with the duration of search defined as an episode $\tau \in \mathcal{E}$ is defined by $\tau = (e, o, x_0^w)$ with $e \in E$, $g \in \mathcal{G}$ and $\mathcal{E}$ denoting the set of navigation episodes. Each episode τ is an object finding task with a fixed episode length T, wherein at each timestep $t \in T$, the robot executes an action a_t from its action space $\mathcal{A}$ to navigate in the environment. $\mathcal{A}$ consists of four actions: a_{left}, a_{right}, $a_{forward}$, a_{stop}. At each timestep t, the robot receives an egocentric visual observation $I_t = (I_t^{\text{rgb}}, I_t^{\text{depth}})$ from a noiseless RGB-D camera, where $I_t^{\text{rgb}} \in \mathbb{R}^{H \times W \times 3}$ for the RGB image and $I_t^{\text{depth}} \in \mathbb{R}^{H \times W}$ for the depth image and pose $\mathbf{x_t^w}$ from a noiseless GPS+Compass sensor. If the robot is within c meters of $\mathbf{x_g^w}$ and g is visible from $\mathbf{x_t^w}$, the robot calls the special action a_{stop}. Episodes terminated by the robot under this criterion are considered successful. An episode is automatically terminated if the robot exceeds the episode length T and is considered to be unsuccessful. A typical ObjectNav task is shown in Figure 3.1 where a robot has to navigate to an object in the environment.

3.2 Open Vocabulary Perception

The ability to detect and reason about the wide range of objects in real-world household environments is crucial for success in ObjectNav. As delibrated in Chapter 2, handling this diversity is challenging for traditional supervised learning methods, which are typically limited by closed-set class definitions, constrained class annotations, and the high costs of manual data labeling. As a result, fully supervised models often struggle to identify and reason about objects outside of predefined categories. To address these limitations, unsupervised or weakly supervised approaches, such as vision language models (VLMs), have gained prominence in the field of visual perception tasks. VLMs are trained on image-text pairs, aiming to maximize the similarity between correct image-text pairs within a shared feature space while minimizing similarity with incorrect pairs. Since large scale internet data used to train VLMs, they contain a broad correlated vocabulary of semantic concepts from images and text in the model's feature space, a capability referred to as "open vocabulary". This enables better generalization to new categories which were not present in the training data.

VLMs have become the go-to approach for improving concept diversity and model generalization across various scene understanding tasks, including classification, object detection, and segmentation. The open vocabulary perception capability of VLMs makes them well-suited for object perception tasks, enabling the recognition of the diverse array of objects that can be present in indoor environments. The success of VLMs is largely attributed to self-supervised training on multimodal image and text data, in which transformers [114, 23] have emerged as

the dominant architecture. State-of-the-art VLMs, such as CLIP [85], are trained on carefully curated high-quality image-text pairs. The CLIP model includes both a text encoder and an image encoder, as shown in Figure 3.2. The text encoder is a standard transformer architecture [114] that converts a text prompt $p_i \in \mathcal{P}$ specified in natural language into a text embedding $l_{p_i} \in \mathbb{R}^{512}$ which represents the text prompt. The image encoder is a vision transformer (ViT) [23] that converts an image $I_i \in \mathcal{I}$ into an image embedding $l_{I_i} \in \mathbb{R}^{512}$. CLIP is trained on a batch of N image-text pairs to predict which of the $N \cdot N$ possible pairings within the batch are correct. The model learns a joint embedding space by training an image encoder and a text encoder together. The goal is to maximize the cosine similarity for the N correct pairs and minimize it for the $N^2 - N$ incorrect pairings using the *contrastive loss*, which is a symmetric cross-entropy loss over the similarity scores. This is equivalent to pulling correct image-text pairs together while pushing incorrect pairs away in the high dimensional embedding space. An example of a correct and incorrect pair is depicted in Figure 3.3.

$$S_i(l_{I_i}, l_{p_i}) = \cos(\theta_{l_i}) = \frac{l_{I_i} \cdot l_{p_i}}{\|l_{I_i}\| \|l_{p_i}\|} \ , \tag{3.1}$$

Cosine similarity between the feature embeddings l_{I_i}, l_{p_i} for a single image-text pair (I_i, p_i) where $I_i \in \mathcal{I}$ and $p_i \in \mathcal{P}$ is defined in Equation (3.1). It is important to note that cosine similarity is bounded in $[-1, 1]$ where a similarity of -1 corresponds to the feature embeddings being antiparallel (semantically irrelevant), 0 corresponds to the embeddings being orthogonal and 1 corresponds to the embeddings being parallel (semantically highly relevant). However, in practice, since VLMs are trained for multi-modal alignment with the objective of maximizing the similarity between matching image-text pairs and minimizing

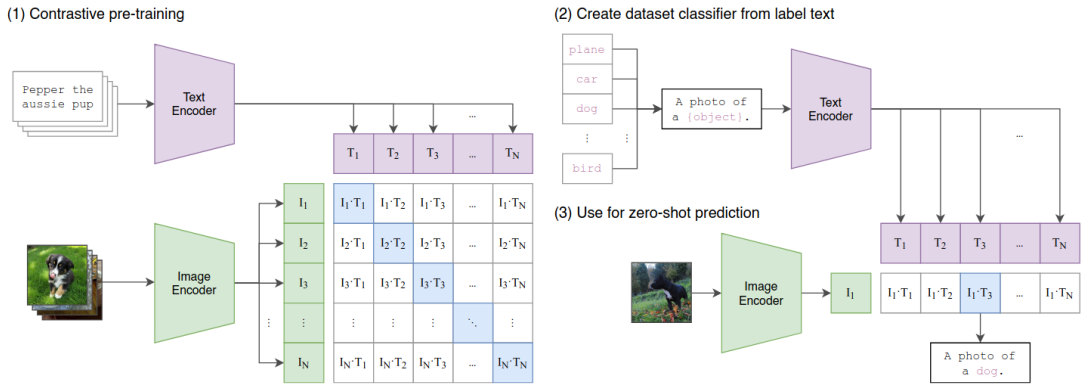


Figure 3.2: CLIP [85] is a vision-language model (VLM). CLIP has an image and text encoder which are jointly trained on an internet-scale dataset of image and text data using contrastive learning (1). CLIP can be used for downstream perception tasks such as image prediction and image-text matching tasks (2,3). Courtesy: *OpenAI*

Figure 3.3: In image-text matching, the goal is to match images with text based on their semantic relevance. In this example, (left) shows a correct match of an image of a kitchen with the text prompt "cappuccino", as the text is semantically relevant to the image. Meanwhile, an incorrect match of the same image is shown with the prompt "bed" (right).

similarity for mismatched pairs, cosine similarity scores of feature embeddings encoded through the model are found to be in the $[0, 1]$ interval for well-aligned VLMs like CLIP.

The process of image-text matching forms a fundamental part of our approach to searching for the target object in ObjectNav, enabling us to evaluate the semantic relevance between the target object we aim to locate and the current image captured by the robot. By calculating the cosine similarity, we can determine the semantic relevance of the observed scene in the purpose of semantically guiding exploration towards the target object. While CLIP exhibits strong performance in image-text matching, it has since been outperformed by more advanced models. At present, the state-of-the-art VLM for image-text matching is BLIP-2 [62].Unlike CLIP, which uses a joint embedding space, BLIP-2 employs separate pre-trained transformers for encoding images and text. For the image encoder, BLIP-2 uses CLIP while for the text encoder, it uses the a FlanT5 model, utilizing them as frozen backbones. To bridge these two disconnected embedding spaces, BLIP-2 trains a transformer called Q-Former.

Given that both CLIP and BLIP-2 have been extensively trained on data from indoor environments, they are well-suited to our application. In our semantic exploration pipeline, we use BLIP-2 to guide the robot toward semantically relevant regions in the observed environment. Cosine similarity from BLIP-2 is used as the metric to quantify the relevance between images captured by the robot and the target object description specified by the user in natural language, leveraging BLIP-2's enhanced capabilities in image-text matching.

The success of vision-language models has also extended their application to object detection tasks. Several approaches have attempted to directly apply CLIP for predicting regions corresponding to objects within images, leveraging knowledge distillation from CLIP to existing object detectors such as RCNNs [38, 131]. However, as these models are trained on image-text pairs, they exhibit poor performance in detecting objects due to domain shifts [133],(Figure 3.4)

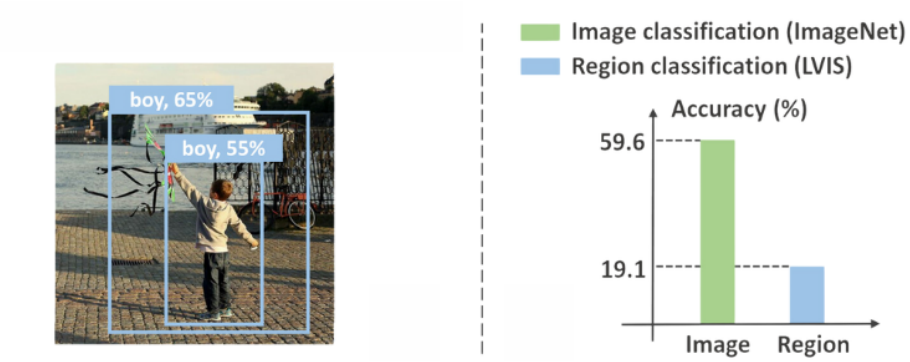


Figure 3.4: The CLIP model is effective at relating images to text, making it suitable for image classification tasks. However, CLIP does not perform as well on region classification, which is essential for object detection. Courtesy: *Zhong et al.* [133]

GLIP [64] introduced the concept of grounded pre-training. Grounding refers to the process of linking or associating abstract linguistic concepts to specific, observable elements in images, such as objects. Grounded pre-training, in contrast to leveraging pre-trained VLMs directly for detection, trains a model with contrastive training on multi scale image-text features which capture such specific elements. These features for images are regions or patches and for text are words and phrases. GLIP uses language phrases like "a person with an orange umbrella" and aims to align such phrases with images containing both nouns. This additional data aids in learning aligned semantics at phrase and region levels. Building upon GLIP's architecture, Grounding DINO [69] combines a transformer-based detection model, DINO [15], with grounded pre-training. When provided with an image-text pair, the model outputs multiple pairs of object boxes and noun phrases. As illustrated in Figure 3.5, the model employs a dual-encoder-single-decoder architecture and operates in five key phases: feature extraction, feature enhancement, language-guided query selection, cross-modality decoding, and bounding box refinement.

In the first phase, feature extraction, text and image features are extracted independently using their respective encoders. In the feature enhancement phase, these features are fused using attention mechanisms [114], allowing visual features to be contextualized with the associated text. The language-guided query selection phase uses textual information to guide query initialization, similar to the process in DINO, and extracts relevant features from the text. A similarity matrix is calculated between image and textual features, allowing the model to focus on the most relevant regions of the image by selecting the top queries based on similarity scores. The final phase, cross-modality decoding, applies attention mechanisms to both image and text features to refine query representations. This improves the accuracy of object detection and classification. In this thesis, addressing the ObjectNav problem requires the robot to detect an arbitrary object

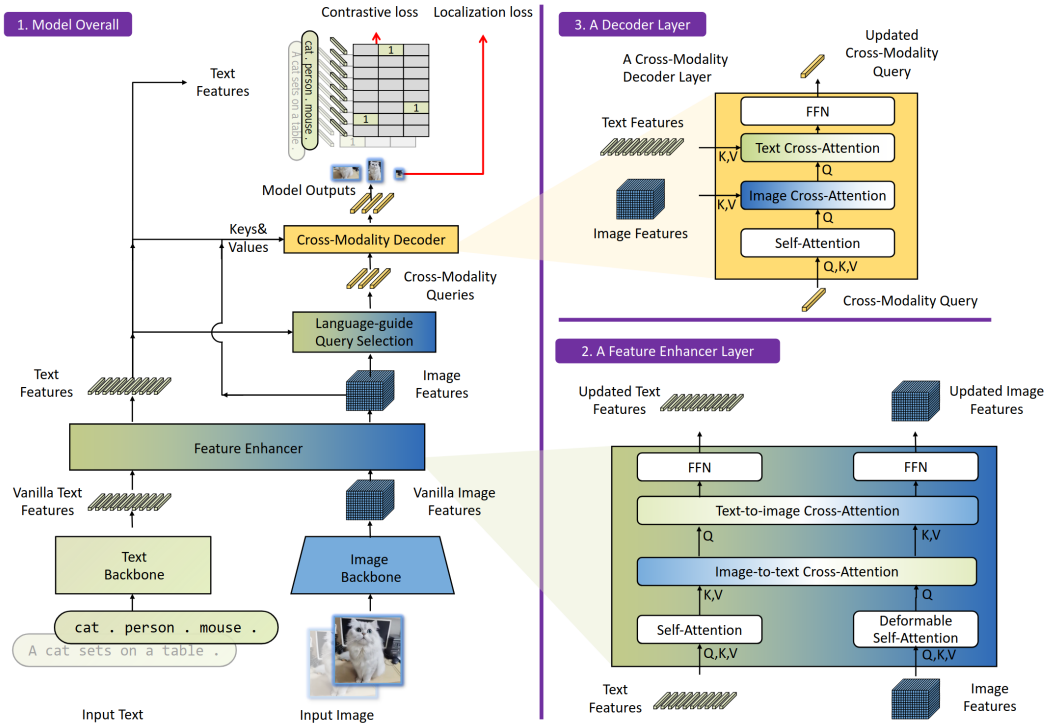


Figure 3.5: Grounding DINO is a transformer-based model designed for open vocabulary object detection. It utilizes vision and text encoders to detect objects based on natural language prompts. The model employs a cross-attention mechanism to fuse image features with text embeddings, enabling precise, prompt-based object localization. Courtesy: *Liu et al.*[69]

based on textual descriptions. The task is executed successfully if the robot detects the object using natural language descriptions and navigates to it. Given Grounding DINO's state-of-the-art performance in open vocabulary object detection, which enables detection beyond predefined object categories, we use it as the open vocabulary object detector within our pipeline.

3.3 Grid Maps

While searching for objects, a robot needs to build and update an internal representation of the environment from its observations. This allows it to track explored areas, identify potential target locations, and plan paths. Grid maps are such a representation which represent information about the environment by discretizing it into so-called grid cells. Each grid cell can store geometric or semantic information about its area. A popular variant of grid maps are occupancy grid maps, which represent occupancy information in 2D, an example of which is illustrated in Figure 3.6. In our active perception framework, we use grid maps extensively to store occupancy and semantic data. In this section, we present the original formulation of grid maps by Moravec and Elfes [76].

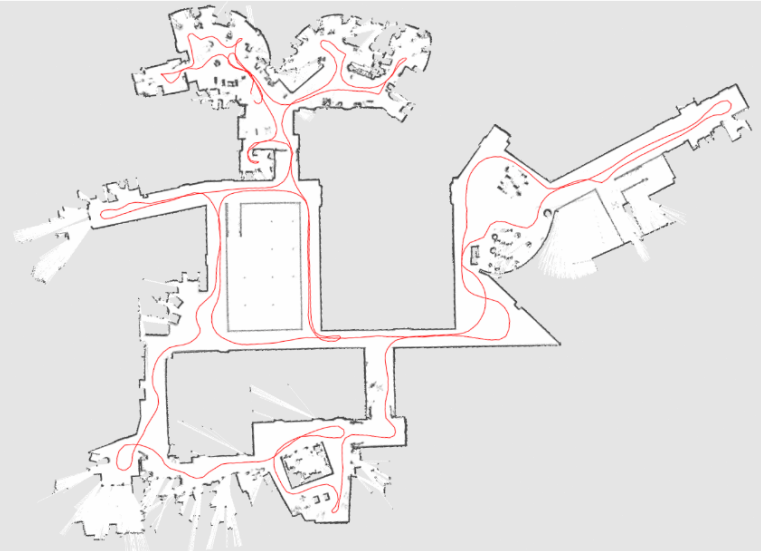


Figure 3.6: Example of an occupancy grid map. Courtesy: *Cyrill Stachniss* [110]

Grid maps create a rigid, equi-distant grid of equally sized cells. We assume that cells are spatially independent of each other and the environment is static. Further, we assume known poses of the robot $x_{1:t}$ where the subscript denotes a sequence of poses until time t. The sequence of sensor measurements $z_{1:t}$ from those poses is collected with noisy sensors. Each grid cell c of a grid map stores a probability $p(c)$ of its state which can signify different contexts depending on the type of map used. For instance, in occupancy grid mapping, it is the probability of the cell being occupied or free whereas in semantic grid mapping, this probability would be the semantic relevance of the cell. These assumptions allow us to formulate the probability of a map m as the product over probabilities of individual cells as:

$$p(m) = \prod_{c \in m} p(c). \tag{3.2}$$

The probability of individual cells $c \in m$ given the measurements $z_{1:t}$ collected by the robot at poses $x_{1:t}$ can be computed using the Bayes rule

$$p(c \mid x_{1:t}, z_{1:t}) \quad = \quad \frac{p(z_t \mid c, x_{1:t-1}, z_{1:t-1})\ p(c \mid x_{1:t}, z_{1:t-1})}{p(z_t \mid x_{1:t}, z_{1:t-1})}. \tag{3.3}$$

The environment is assumed to be Markovian. i.e z_t does not depend on $x_{1:t-1}$ and $z_{1:t-1}$, leading to

$$p(c \mid x_{1:t}, z_{1:t}) \quad = \quad \frac{p(z_t \mid c, x_t)\ p(c \mid x_{1:t}, z_{1:t-1})}{p(z_t \mid x_{1:t}, z_{1:t-1})}. \tag{3.4}$$

We apply Bayes' rule for the term $p(z_t \mid c, x_t)$ in Equation (3.4) and obtain

$$p(z_t \mid c, x_t) = \frac{p(c \mid x_t, z_t)\ p(z_t \mid x_t)}{p(c \mid x_t)}. \tag{3.5}$$

We can combine Equation (3.5) and Equation (3.4). In addition to that we can assume that x_t does not carry any information about c if there is no observation z_t. i.e., the state of the map changes only when there is an observation from the robot. Following this assumption, we can write

$$p(c \mid x_{1:t}, z_{1:t}) = \frac{p(c \mid x_t, z_t)\ p(z_t \mid x_t)\ p(c \mid x_{1:t-1}, z_{1:t-1})}{p(c)\ p(z_t \mid x_{1:t}, z_{1:t-1})}. \tag{3.6}$$

In an occupancy grid map, each cell of the environment is assumed to be in only one of two possible states: free or occupied. We therefore make use of complement and estimate the probability of cell c being in an opposite state to the one estimated in Equation (3.6)

$$p(\neg c \mid x_{1:t}, z_{1:t}) = \frac{p(\neg c \mid x_t, z_t)\ p(z_t \mid x_t)\ p(\neg c \mid x_{1:t-1}, z_{1:t-1})}{p(\neg c)\ p(z_t \mid x_{1:t}, z_{1:t-1})}. \tag{3.7}$$

We divide Equation (3.6) by Equation (3.7) and obtain

$$\frac{p(c \mid x_{1:t}, z_{1:t})}{p(\neg c \mid x_{1:t}, z_{1:t})} = \frac{p(c \mid x_t, z_t)\ p(\neg c)\ p(z_t \mid x_{1:t}, z_{1:t-1})}{p(\neg c \mid x_t, z_t)\ p(c)\ p(z_t \mid x_{1:t}, z_{1:t-1})}. \tag{3.8}$$

Finally, we use the fact that $p(\neg c) = 1 - p(c)$ which yields

$$\frac{p(c \mid x_{1:t}, z_{1:t})}{1 - p(c \mid x_{1:t}, z_{1:t})} = \frac{p(c \mid x_t, z_t)}{1 - p(c \mid x_t, z_t)} \cdot \frac{1 - p(c)}{p(c)} \cdot \frac{p(c \mid x_{1:t-1}, z_{1:t-1})}{1 - p(c \mid x_{1:t-1}, z_{1:t-1})}. \tag{3.9}$$

Given all of the above equations, we can specify the full *occupancy update formula* as follows:

$$p(c \mid x_{1:t}, z_{1:t}) = \left[1 + \frac{1 - p(c \mid x_t, z_t)}{p(c \mid x_t, z_t)} \cdot \frac{p(c)}{1 - p(c)} \cdot \frac{1 - p(c \mid x_{1:t-1}, z_{1:t-1})}{p(c \mid x_{1:t-1}, z_{1:t-1})}\right]^{-1}. \tag{3.10}$$

Equation (3.10) tells us how to update our belief $p(c \mid x_{1:t}, z_{1:t})$ about the occupancy probability of a grid cell given sensory input. In practice, one often assumes that the occupancy prior is 0.5 for all cells, so that $\frac{p(c)}{1-p(c)}$ can be removed from the equation.

In this section, we will skip the computation of the occupancy probability $p(c \mid x_t, z_t)$ of a grid cell given a *single* observation z_t and the corresponding pose

x_t of the robot. This quantity depends on the sensor of the robot and has to be defined manually for each type of sensor. For more information on grid maps we refer the reader to the original version of the Moravec and Elfes paper [77] and the well written theses on this topic [110, 9].

In our approach for ObjectNav, we maintain two maps. We utilize the standard occupancy grid mapping approach with known poses to maintain an obstacle map of the environment. In addition, for the second map, we extend the binary occupancy grid map approach to store continuous values of cosine similarity from vision language models and update them using a Bayesian formulation.

3.4 Planning with Monte Carlo Tree Search

Path planning is a fundamental component of any robot navigation pipeline and is one of the major focuses of this thesis. Robotic path planning can be formulated as a sequential decision-making problem where at each step, a decision has to be made about where to go next. If a robot has a map of the environment and uses it to plan, the environment is termed as being "fully observable". Such sequential decision-making problems in fully observable environments can be modeled as a Markov Decision Process. A Markov Decision Process (MDP) [98] has four components:

- $\mathcal{S}$: A set of states, with the initial state defined as s_0.
- $\mathcal{A}$: A set of actions, with an action selected from this set as $a \in \mathcal{A}$
- $T(s, a, s')$: A transition model, which defines the probability of arriving at state s' if action a is taken in state s
- $R(s, a, s')$: A reward function

Decisions are modeled as state-action pairs in which each next state s' depends on the current state s and action taken a. A policy is a mapping from states to actions, specifying the action that would be taken from each state in S. Finding the optimal policy π that yields the *maximum expected reward* is the goal of decision-making in problems formulated as a MDP. Among the many approaches to solve MDPs, Monte Carlo Tree Search (MCTS) is a simple and powerful approach for determining the optimal policy in complex planning problems. MCTS has had a profound impact on decision-making and planning problems, such as building bots for strategy games like Chess and Go [107], planning and other decision-making problems. This capability extends to applications like robot navigation. MCTS is based on two key principles. First, the true value of an action can

Algorithm 1 Monte Carlo Tree Search (MCTS)

Input: Root node s_0
Output: Action from root node
while iterations not finished **do**
  $v \leftarrow$ **Select**(s_0) ▷ Selection phase
  if v is not a terminal state **then**
    $v' \leftarrow$ **Expand**(v) ▷ Expansion phase
    $\Delta \leftarrow$ **Simulate**(v') ▷ Simulation phase
  else
    $\Delta \leftarrow$ **Simulate**(v) ▷ Simulation from the terminal state
  Backpropagate(v, Δ) ▷ Backpropagation phase
return best child of s_0 based on visit count

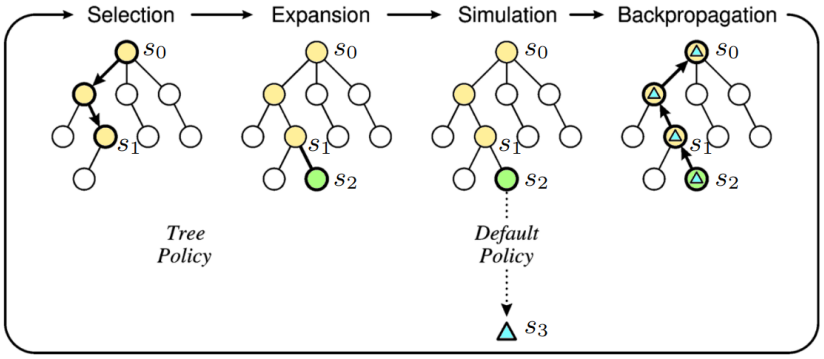


Figure 3.7: A step-by-step visual guide to the four phases of MCTS: selection, expansion, simulation, and backpropagation. Starting at the root, the algorithm selects child nodes using the tree policy, expands by adding new moves as nodes, simulates outcomes through random rollouts, and backpropagates results back up the tree. Courtesy: *Duguépéroux et al.* [26]

be approximated through simulations. Second, these approximations can be efficiently leveraged to adjust the policy towards a best-first strategy [12]. The algorithm incrementally constructs a search tree by randomly sampling from the decision space, guided by the outcomes of previous explorations within the tree. This tree serves as a tool for estimating the value of different moves, with the estimates, especially for the most promising moves, becoming more accurate as the tree grows. The MCTS algorithm can be divided into four key phases:

- *Selection*: From the root node, a tree policy is recursively applied to traverse the tree. The tree policy selects the child node that maximizes the expected cumulative reward.

- *Expansion*: When leaf node is reached, one or more child nodes are added to the tree, according to the available actions.

- *Simulation*: A Monte-Carlo simulation is run by sampling random actions from the given non-terminal node to a terminal state to produce a value estimate. This is also known as the rollout policy.

- *Backpropagation*: The results of the simulation are propagated back up the tree through the selected nodes to update their statistics.

These steps are summarized as pseudocode in Algorithm 1. At each selection step in the tree search, one is faced with a dilemma, to choose an action $a \in A$ in order to maximize the expected cumulative reward by consistently taking the optimal action. Since the distribution of rewards over the planning horizon is unknown, potential rewards must be estimated based on collected observations. This is the so-called "exploration-exploitation dilemma". The MCTS needs to balance the exploitation of the action currently believed to be optimal, i.e, yields the highest immediate reward, with the exploration of other actions that currently appear sub-optimal but may be superior in the long run. In MCTS, the balance between exploration and exploitation is achieved using a tree policy approximating the true reward of actions that can be taken from the current state. One possible tree policy is the Upper Confidence Bound for Trees (UCT) developed by Kocsis and Szepesvári [55]. In deciding which child node to select, the value of each child node is estimated through multiple iterations i of Monte Carlo simulations and yields rewards $R_{i,1}, R_{i,2}, \ldots$, which are independently and identically distributed. Hence, these rewards can be considered as corresponding to random variables with unknown distributions having an unknown expectation Q_i. The cumulative sum of these expected rewards over the planning horizon is the quantity to be maximized by the tree search policy. In UCT, a child node i is selected to maximize:

$$UCT(i) \quad = \quad \frac{Q_i}{n_i} + C_e \cdot \sqrt{\frac{\ln N}{n_i}} \, , \tag{3.11}$$

where N is the number of times the parent node of the selected child has been visited, n_i the number of times child i has been visited and $C_e > 0$ is known as the exploration constant that is used to control the amount of exploration. There is a crucial balance between the first (exploitation) and second (exploration) terms in the UCB equation. With each visit to a node, the denominator of the exploration term increases, reducing its contribution to the overall value. Conversely, when another child of the parent node is visited, the numerator in the exploration term increases, which raises the exploration values of the unvisited sibling nodes. The algorithm also ensures that all children of a node are evaluated at least once before expansion, as a visit count of $n_i = 0$ results in a UCT value of ∞. Setting the value of the exploration constant in UCT is non-trivial for arbitrary rewards.

For rewards in the range $[0, 1]$, Kocsis and Szepesvári suggest that the value of $\frac{1}{\sqrt{2}}$ should be used.

MCTS is a simple, scalable, and domain-independent algorithm for sequential decision-making that can be flexibly adapted to various problems formulated as MDPs. However, the algorithm has some drawbacks, particularly its high memory demands and the large number of iterations required to converge to an optimal policy. These factors can make MCTS challenging to implement in time-sensitive applications. Nonetheless, MCTS is an anytime algorithm, meaning that increased computational power generally results in better performance. Since robot navigation in household environments is not as time-critical as domains like autonomous driving, where split-second decisions are crucial to avoid severe damage, we deem it suitable to use MCTS for our problem. We adapt the MCTS algorithm with UCT tree policies to build the path planner in our ObjectNav approach. As a non-myopic method, MCTS generally performs better than greedy planning strategies, especially in exploration tasks where future rewards need to be considered for effective navigation. To ensure compute-efficient implementation, problem-specific modifications and integration with our grid mapping system were made, allowing the planner to function effectively within our pipeline.

Chapter 4

Approach

In this chapter, we introduce our approach to the ObjectNav problem. Our approach involves modular components for perception, mapping, exploration, and planning. Our method aims to equip a robot with the capability to find a user-specified object in natural language (e.g., chair, wallet, clothes) within an unknown 3D environment. This natural language description of the object to find, is referred to as the "target object". Given the uncertainty of the initially unknown environment, exploration becomes essential, as the robot must actively search through uncharted spaces to identify areas most likely to contain the target object. In our pipeline, we assume that the robot is equipped with an RGB-D camera and pose information provided by a GPS+Compass sensor. The sensors are assumed to be noise-free, in accordance to the assumptions of the ObjectNav problem defined in Section 3.1. As illustrated in Figure 4.1, our pipeline consists of two modules: "object detection" and "active semantic exploration". At each time step, images from the camera are processed through the object detection module to detect the target object within the current RGB image. If detected, the robot directly navigates to the object using point-goal navigation, as outlined in Section 4.1. However, when the object is not detected given the current sensor observations, the system transitions to the active semantic exploration module, which constitutes the main contribution of this thesis.

Our exploration strategy is designed to systematically search unknown environments for a target object by performing goal-directed exploration guided towards *semantically relevant* regions in the environment. We define a semantically relevant region as an area with a high likelihood of containing the target object. For instance, if the target object is a "cup," then a semantically relevant region in a household might be the kitchen. In order to estimate semantic relevance, we treat VLMs as a deep learning-based sensor which performs image-text matching between the images captured by the robot and the target object description. Realizing the inherent uncertainty in estimating semantic relevance from

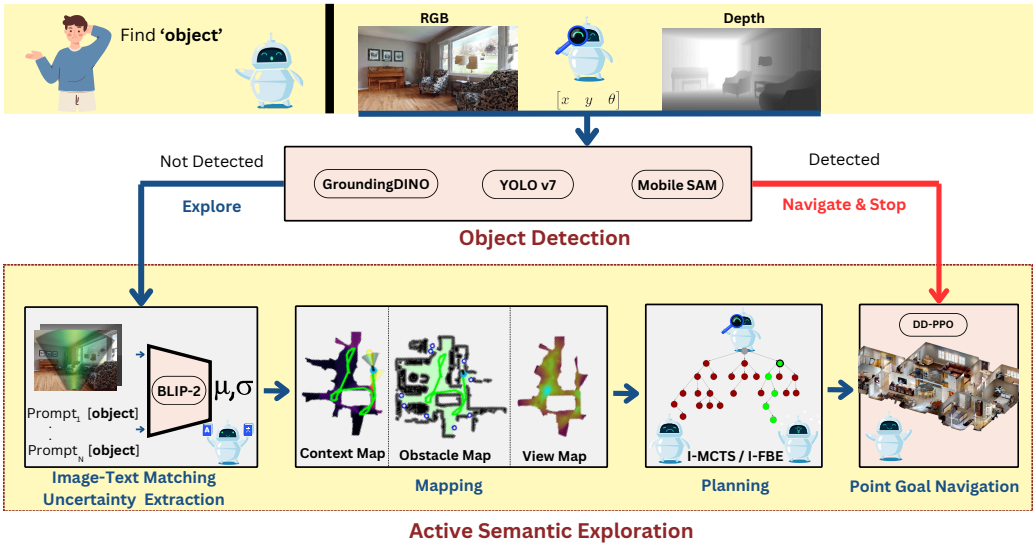


Figure 4.1: Our ObjectNav approach consists of two main components: object detection and active semantic exploration. If the target object is detected in the robot's immediate frame, the robot navigates directly to it. Otherwise, it explores the environment using a planning strategy guided by an uncertainty-aware context map. This map incorporates semantic cues and their associated uncertainty, derived from image-text matching, to relate the robot's observations to the target object effectively.

VLMs, we develop a probabilistic sensor model, detailed in Section 4.2. Since VLMs take both an image and text as input, they estimate the likelihood of an object's presence in an image.

As different images may cover the same or overlapping regions, we construct a metric-semantic representation of the environment, or "context map", to integrate semantic relevance observed from various viewpoints captured over multiple time steps and continuously updated through probabilistic updates. The context map is defined as $M_C : m \to [0,1]^{H \times W}$, over a grid lattice m with $H \times W$ spatially independent cells, each cell of the map represents semantic relevance, which we model as a Gaussian distribution, as explained in Section 4.3. In addition, we develop an "obstacle map", and a "view map". The obstacle map M_O is an occupancy grid map representation used to differentiate between free and occupied regions in the environment. It is continuously updated with sensor data to reflect changes in the environment and successively updated using probabilistic updates, as previously explained in Section 3.3. The obstacle map $M_O : m \to \{0,1\}^{H \times W}$, is also defined over a grid lattice m with $H \times W$ spatially independent cells. Our specific occupancy grid map representation is a map of occupancy probabilities created from depth camera images and closely follows the approach of [128]. Additionally, the view map $M_V \to \mathbf{N}^{H \times W}$, is maintained to track the frequency of regions observed in the environment and is non-probabilistic. This framework

Figure 4.2: We incorporate YOLOv7 [116] and Grounding DINO [69] object detection models for enabling object detection in our pipeline. MobileSAM is used to segment the target object in the depth camera images. This example shows YOLOv7 detecting the target object "chair" in the RGB image (left) and MobileSAM segmenting the detected objects in the depth frame to determine a waypoint for reaching the object (right).

enables the robot to assess contextual relevance across regions, prioritizing exploration in areas with a higher probability of finding the target object while accounting for uncertainty.

We develop two planning algorithms, informative frontier-based exploration (I-FBE) and informative monte carlo tree search (I-MCTS) which explore the environment based on our maps, as described in Section 4.4.2. Both planners serve as high-level strategies that balance the trade-off between navigating to high relevance regions on the context map (exploitation) and exploring high variance regions. Point-goal navigation acts as a low-level planner to reach these regions. This approach continuously refines the robot's belief in the semantic relevance and associated uncertainty of the environment, guiding its actions based on semantic cues for uncertainty-aware exploration.

4.1 Object Detection and Point Goal Navigation

In order to succeed in the ObjectNav task, it is crucial for a robot initialized in an environment $e \in E$ at pose $\mathbf{x}_0^w$ to detect the goal object g specified by the natural language description o and navigate to within c meters its pose $\mathbf{x}_g^w$. This necessitates an object detection subsystem in our pipeline. We follow the approach of state-of-the-art works [128] to develop the architecture of our object detection module with the assumption that the robot has access to noiseless RGB and depth images (I_t^{rgb}, I_t^{depth}) from its egocentric RGB-D camera feed as well as its pose $\mathbf{x}_g^w$. Each RGB image frame received by the robot is processed through our object detection module, which attempts to locate the specified object in the frame. This module uses a hybrid combination of two object detectors. *YOLOv7* [116] is

a supervised learning-based detector trained on the MS-COCO [66] dataset for objects that pertain to the categories of that dataset. *Grounding DINO* [69] is an open vocabulary object detector capable of detecting a wide range of objects specified by arbitrary text prompts, beyond a fixed set of predefined classes. This combination is due to the higher accuracy of YOLOv7 for object classes contained in MS-COCO. Once the target object is detected in the frame, we use the segmentation model *Mobile SAM* to segment the object in the depth image. Using the segmented part of the image, we determine a waypoint on the object for the robot to navigate towards. To navigate to this waypoint from the current location of the robot, we utilize the state-of-the-art point-goal navigation approach, *DD-PPO* [118], which is trained on a vast variety of indoor scenes using reinforcement learning and requires I_t^{depth} and $\mathbf{x}_t^w$ to navigate to the waypoint.

4.2 Sensor Model for Semantic Relevance

When humans search for an object in a new environment, they employ a semantically guided strategy. Rather than exploring all areas equally, they prioritize regions where the target object is more likely to be found. This prioritization is based on prior knowledge that links the *semantic relevance* of observed regions, which reflects how contextually appropriate a location is for finding a specific object, to the object being sought. This usage of prior knowledge directs the search toward the areas with the highest likelihood of success, aligning with the principles of the theory of active perception [2, 80]. For example, when searching for an apple, a person is more likely to explore a semantically relevant area, such as the kitchen, while de-prioritizing less relevant locations, such as the bathroom.

Inspired by human behavior, recent ObjectNav approaches have utilized VLMs to enable semantically informed exploration [128, 30, 52] by performing image-text matching. As mentioned in Section 3.2, VLMs have the ability to associate contextual information from both text and images, positioning them as a powerful tool for estimating semantic relevance. VLMs calculate cosine similarity S_i between an image I_i and a text prompt p_i which serve as a realization of this semantic relevance. However, standardized methods for evaluating their level of contextual comprehension, remain underdeveloped. While VLMs can encode contextually rich associations, the internal representation of semantic concepts within their transformer-based text and image encoders are largely opaque. This limitation of interpretability underscores their "black box" nature. The architecture of VLMs, which typically relies on transformer-based encoders, exhibits high sensitivity to minor variations in input, making consistent performance across diverse tasks challenging to guarantee. This also reflects that there is inherent uncertainty in estimating the semantic relevance for a region in the environment

through VLMs, as different text prompts, even if contextually the same, can lead to different cosine similarities. Similarly, viewpoints changes during image acquisition or lighting conditions can result in different cosine similarities.

This motivates us to treat cosine similarity $S_i(I_i, P_i)$ as an uncertain sensor measurement of the true semantic relevance S of some region in space in the context of the target object as specified by o. We develop an uncertainty quantification scheme in the following subsections, we formalize the uncertainty in image-text matching in Section 4.2.1. Further, we show how uncertainty could be quantified for context due to prompts in Section 4.2.2 and due to viewpoint changes, as detailed in Section 4.2.3.

4.2.1 Uncertainty in Image-Text Matching

As detailed in Section 3.2, VLMs can perform image-text matching, i.e, they can quantify the semantic relevance of an image $I_i \in \mathcal{I}$ and a text prompt $p_i \in \mathcal{P}$ by computing the cosine similarity S_i between their high dimensional feature embeddings l_{I_i} and l_{p_i} respectively, which is deterministic. However, research suggests that there is a semantic gap between visual and textual representations in VLMs. The semantic gap occurs due to the fact that there can be multiple prompts $p_i \in \hat{\mathcal{P}}$ which may be linguistically different but convey the same context with respect to I_i. Here, $\hat{\mathcal{P}}$ denotes the set of prompts relevant to I_i, with $\hat{\mathcal{P}} \subsetneq \mathcal{P}$, where $\mathcal{P}$ is the infinite set of all possible natural language descriptions. The semantic gap introduces uncertainty in estimating the true semantic relevance S of an image and it can only be estimated by realizations of cosine similarities S_i for different prompts, even if these prompts convey the same context. We argue that due to the semantic gap, cosine similarity can't be treated as a deterministic function for semantic relevance and is only an uncertain realization of the true semantic relevance between I_i and p_i. Our proposed exploration pipeline expands on this concept by quantifying the noise or uncertainties in these realizations employing VLMs to estimate the semantic relevance of images captured by the robot in household environments to a target object. This process assesses the potential utility of the observed area for locating the object, facilitating more efficient and context-aware exploration of the environment.

There has been a growing trend in the use of VLMs in robotics [30, 128], however research examining the impact of data uncertainties such as variations in prompt phrasing on robotics tasks remains sparse. In the context of the ObjectNav problem, VLFM [128] is one of the first studies to leverage cosine similarities from VLMs for semantic navigation. This approach employs the BLIP-2 model for image-text matching using the current RGB frame and uses a fixed prompt format "Seems like there is a `target_object` ahead", where the `target_object` is specified based on user requests. BLIP-2 is impacted by the issues of data

uncertainty, as evidenced by empirical studies demonstrating that prompt selection significantly influences model outputs [20, 136, 132]. Given the scale of such tasks like ObjectNav where numerous object types and spatial variations must be accounted for, prompt tuning becomes impractical. Additionally, optimizing prompts on a test dataset may risk overfitting, thereby limiting generalizability to unfamiliar environments. Motivated by these challenges, we investigate data uncertainty within VLMs by examining how variations in prompts affect the encoded context or cosine similarity metrics. We propose modeling this variability as a random variable in a probabilistic framework, aiming to leverage this uncertainty to enhance navigation performance. Subtle shifts in prompt wording, image perspectives, or image augmentations can cause substantial fluctuations in performance due to the associations learned during training. This sensitivity to input perturbations has been extensively documented in the literature [136, 135, 104, 120, 53] which highlight the challenges of achieving consistent performance across varied contexts. As a result, prompt engineering which is tuning word choices to optimize output, remains a common but time-intensive practice. Nonetheless, prompt engineering is typically task-specific, aiming to maximize model performance for a narrow focus. Extending this approach to numerous general downstream tasks, particularly in robotics, becomes infeasible due to the substantial time and computational resources required.

4.2.2 Extracting Context Uncertainty

Given two vectors l_{I_i} and l_{p_i} representing, image and text embeddings for image $I_i \in \mathcal{I}$ and $p_i \in \mathcal{P}$ respectively. The cosine similarity $S(l_{I_i}, l_{p_i})$ is calculated as previously defined in Equation (3.1):

$$S_i(l_{I_i}, l_{p_i}) = \cos(\theta_{l_i}) = \frac{l_{I_i} \cdot l_{p_i}}{\|l_{I_i}\| \|l_{p_i}\|} \ ,$$

where θ_l represents the angle between I_i and p_i in the high-dimensional feature space. However, due to the limited interpretability of VLMs' internal mechanisms, the exact structure of this space remains difficult to understand, and cosine similarity scores exhibit variability, due to factors such as prompt variations. To capture this variability, we model semantic relevance as a random variable S with a probability distribution that reflects the uncertainty in cosine similarity interpretations. For instance, if we assume S follows a Gaussian distribution, we can express it as:

$$P(S) \ \sim \ \mathcal{N}(\mu, \sigma^2) \ , \tag{4.1}$$

where μ is the mean similarity score and σ^2 is the variance. The probability density function (PDF) of this Gaussian distribution is given by:

$$PDF(S_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(S_i - \mu)^2}{2\sigma^2}\right) . \tag{4.2}$$

This formula captures the likelihood of different cosine similarity scores, reflecting the mean and variance of semantic similarity values. This probabilistic treatment of S enables cosine similarity to function not merely as a metric but as an indirect measure of likelihood, steering exploration toward regions with higher semantic relevance. By formalizing similarity as a random variable, we encapsulate the uncertainty in semantic interpretations, thus enabling cosine similarity to guide exploration probabilistically rather than deterministically. This approach enhances VLM-driven exploration by focusing on regions with statistically significant semantic relevance, which can then be used to update a probabilistic map which encodes semantic relevance, referred to as context map.

To construct a robust likelihood model, we need a set of cosine similarity estimates across different prompt variations. To achieve this, we use the large language model GPT-4 [78] to generate diverse prompts that capture the target context. We provide GPT-4 with a few initial examples that vary in phrasing but convey consistent context, such as "There is a `target_object` in the vicinity" and "A `target_object` could be ahead". GPT-4 then generates multiple similar prompts to explore different expressions of the target context. This approach allows us to systematically evaluate the impact of prompt variations on performance of VLFM on ObjectNav, as presented in Section 5.2.1. Our approach employs an ensemble of five selected prompts from this set, allowing us to capture a range of semantic interpretations. By calculating the mean and variance of cosine similarity scores for each prompt with the current image frame, we gain insights into the variability of S. Formally, we define the prompt ensemble as a set of prompts $\mathcal{P}$ which consists of N unique prompts $p_1, p_2, \ldots, p_N \in \mathcal{P}$. We denote the image from the robot at the current timestep t as I_t^{rgb}. Both the image and each prompt are encoded through the VLM, resulting in corresponding embeddings $l_{I_t^{\text{rgb}}}$ for the image and l_{p_i} for each prompt $p_i \in \mathcal{P}$. The cosine similarity between the image embedding and a prompt embedding is denoted by S_i, which for N prompts would result in N cosine similarities. This is shown for the image and a single prompt p_i at timestep t as:

$$S_i(l_{I_t^{\text{rgb}}}, l_{p_i}) = \frac{l_{I_t^{\text{rgb}}} \cdot l_{p_i}}{\|l_{I_t^{\text{rgb}}}\| \|l_{p_i}\|} . \tag{4.3}$$

The *mean* μ of of S can be approximated as a mean of the cosine similarity scores:

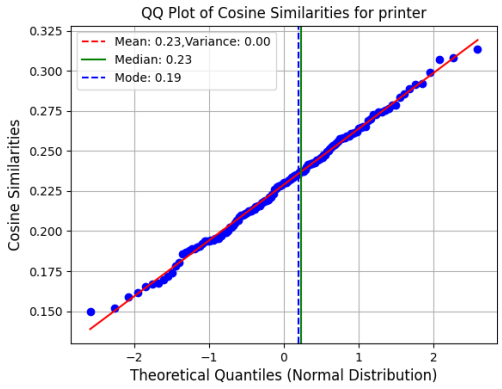


Figure 4.3: In this example, we show that the Quantile-Quantile (QQ) plot of 100 cosine similarity values which are generated from 100 unique prompts around the target object name "printer", lie on a Gaussian distribution.

$$\hat{\mu_S} \quad = \quad \frac{1}{N}\sum_{i=1}^{N} S_i \ . \tag{4.4}$$

The *variance* σ^2 of S can be approximated as a variance $\hat{\sigma^2}$ of the cosine similarity scores:

$$\hat{\sigma_S^2} = \frac{1}{N}\sum_{i=1}^{N}(S_i - \hat{\mu})^2 \ . \tag{4.5}$$

Thus, Equation (4.4) and Equation (4.5) can be used to approximate the random variable $S \approx \hat{S}$ as $P(\hat{S}) \sim \mathcal{N}(\hat{\mu}, \hat{\sigma^2})$ calculated from the prompt ensemble.

To further support our Gaussian assumption for semantic relevance, we analyzed the cosine similarity from the BLIP-2 VLM across various rgb images from a dataset. We used prompt ensembles of sizes ranging from 10 to 100. This approach allowed us to assess the consistency of similarity distributions under prompt variation. For example, when analyzing an office image without a printer, we tested prompts such as "is there a printer ahead" or "printer in the vicinity." Our analysis, as shown in Figure 4.3, indicates that cosine similarity values, when aggregated over prompt ensembles, approximate a Gaussian distribution. This result validates that $\hat{S}$ is normally distributed and can be used to compute posterior probability for the subsequent context map creation process that we present in Section 4.3.

4.2.3 Extracting Viewpoint Uncertainty

In robotic perception, accurately representing viewpoint uncertainty is essential for effective decision-making, as different parts of the robot's field of view con-

tribute unevenly to semantic relevance. When a VLM generates a cosine similarity score for a given text and image pair, this score reflects the semantic relevance across the entire field of view (FOV). However, treating all regions equally can lead to inaccuracies, as some areas may be more relevant to the target than others. Viewpoint uncertainty arises because the robot's view on objects or regions differs in clarity, with objects in the central area along the optical axis being fully observed and peripheral regions may be unclear or only partially visible.

For instance, if the goal is to find a cup and the region along the optical axis shows an unrelated object, such as a toilet, the similarity score for the entire FOV may be lowered. This could cause the system to overlook potentially relevant areas at the peripheries, such as the entrance to a kitchen that might offer clues about the target's location, as illustrated in Figure 4.4. If in the direct line of sight is an unrelated region, then even if semantically related regions are viewed near the periphery, they tend to receive lower scores due to reduced clarity, partial views and less direct alignment with the robot's viewpoint. While the same regions, when seen directly along the optical axis, are assigned higher scores because of increased clarity and semantic confidence while areas along the optical axis. As a result, uncertainty rises as the object shifts from the optical axis to the edges of the FOV, reflecting decreasing confidence in observations.

Current methods, such as VLFM, adjust cosine similarity values based on a pixel's distance from the optical axis, as shown in Equation (4.6). With higher confidence for pixels along the optical axis and lower confidence for peripheral pixels. "The confidence of a pixel in the robot's FOV is determined based on its location relative to the optical axis. Where, θ is the angle between the pixel and the optical axis and θ_{fov} is the horizontal FOV of the robot's camera" [128]. For an image I^{rgb}, it is defined as:

$$C(\theta) = \cos^2\left(\frac{2 \cdot \theta \cdot \pi}{\theta_{\text{fov}}}\right) , \tag{4.6}$$

where θ is the angle between the pixel and the optical axis, and θ_{fov} is the horizontal FOV of the robot's camera.

However, in the case of VLFM, this adjustment is only applied when a grid cell has a value from a previous observation, potentially leaving newly observed areas with inaccurate confidence levels. Additionally, their approach includes a decision threshold that prevents updates if the confidence is too low. These simplifications can affect map quality and reduce the effectiveness of exploration. To overcome these limitations, we propose an approach that applies confidence adjustments consistently, regardless of whether a region has been observed previously. Instead of directly using the confidence score, we use it as a measure of variance, treating viewpoint uncertainty as the opposite of confidence. This approach assigns a variance close to 0 for regions along the optical axis (high confidence) and a

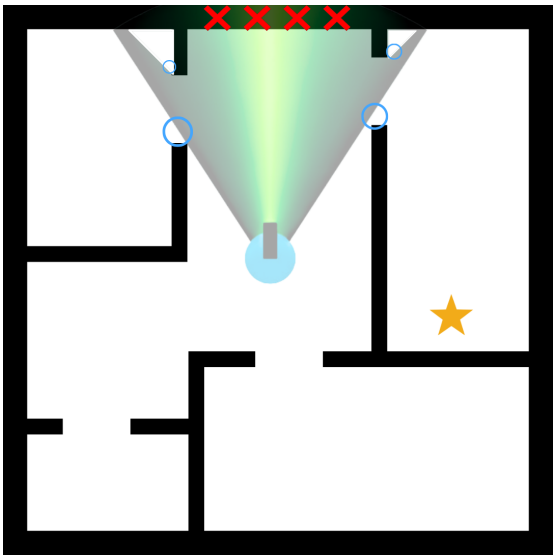


Figure 4.4: In this example, the robot needs to find the goal abject represented by a yellow star. To go towards the object, it needs to explore frontiers. However, the current field of view (FOV) looks at an object (denoted by red crosses) that is dissimilar to the target object. If equal weighting was used, all of the FOV gets the same cosine similarity assigned. However, in our approach, we will have a higher variance for the edges of the FOV, which would increase the uncertainty for those cells on the map.

variance of 1 for regions at the periphery (low confidence). This allows us to incorporate viewpoint uncertainty into the sensor model as:

$$\sigma_v^2(\theta) = 1 - C(\theta) \ , \tag{4.7}$$

where σ_v^2 is the viewpoint uncertainty. While this function is not truly a variance in the statistical sense, as it does not satisfy the mathematical properties of the second moment, we incorporate it in our framework to show the variability of viewpoints as an additional uncertainty in the estimation of S. We integrate the two sources of uncertainties by adding the viewpoint uncertainty as an independent variance into our context uncertainty formulation from Equation (4.5). We combine these within a probabilistic framework. Viewpoint uncertainty is then incorporated as an additional variance component. The total uncertainty for each observation is given by:

$$\hat{\sigma_S^2} = \hat{\sigma_S^2} + \sigma_v^2(\theta) \ , \tag{4.8}$$

where $\hat{\sigma_S}^2$ (left) is redefined as the total variance of the estimated semantic relevance $\hat{S}$ due to the combination of context and viewpoint uncertainties to simplify notation. Here, $\hat{\sigma_S}^2$ (right) represents the variance due to context uncertainty and σ_v^2 accounts for viewpoint uncertainty. $\hat{\sigma_S}^2$ captures the uncertainty

in semantic relevance from each image observation I^{rgb} and is used to update the probability distribution of each grid cell in the our context map, resulting in a nuanced representation that reflects the reliability of the semantic relevance of observed regions within the robot's environment. This map will guide the robot's exploration by highlighting regions with higher potential for locating the target object based on the combined semantic and uncertainty information. By accounting for both context and viewpoint uncertainty during the mapping process, we aim to create more accurate and adaptive map updates that better inform the robot's decision-making as it navigates and explores.

4.3 Uncertainty-Infused Context Map

Traditional grid-based mapping techniques, such as occupancy grid maps, represent each grid cell as a binary random variable, indicating whether it is occupied, free, or unknown. These approaches utilize a prior probability distribution for each cell and apply probabilistic updates to iteratively refine the map's representation of the environment. Beyond spatial data, grid maps can incorporate semantic information, such as features extracted from visual observations. Despite incorporating semantic data, VLFM lacks a probabilistic framework for map updates. Instead, it maintains a point estimate of cosine similarities over the grid cells. Unlike probabilistic approaches, there is no prior probability defined over the grid cells. However if a particular grid cell already has a cosine similarity value associated with it from a previous observation and the grid cells are observed again and receive a new cosine similarity value, they are updated via weighted averaging. Additionally, the approach uses hand-tuned decision thresholds to restrict updates, such as discarding new observations that fall below a specified confidence level. These limitations lead to missed opportunities for incorporating valuable information and hinder the system's ability to rigorously account for uncertainty in the mapping process.

In contrast, a probabilistic mapping framework offers a more theoretically principled approach by explicitly incorporating uncertainty into the mapping process. This allows the map to reflect the reliability of prior data and its uncertainty, leading to a more accurate representation of the environment without using any hand-tuned parameters like the decision threshold of VLFM. To address these limitations, we propose an uncertainty-infused context map that integrates both context and viewpoint uncertainties. This approach enables a simpler and more comprehensive mapping framework, capturing semantic relevance and its associated uncertainty within a probabilistic framework.

Building on these insights, we propose a metric-semantic probabilistic map representation that incorporates contextual information using Bayesian updates

called context map. $M_C : m \rightarrow [0,1]^{H \times W}$. In our context map M_C, each grid cell $c \in M_c$ is treated as a random variable, consistent with the principles of probabilistic grid mapping, as previously discussed in Section 3.3. Our map representation models each grid cell as a random variable that reflects the semantic relevance S , bounded in the interval $[0,1]$. This reflects the range of both expected semantic relevance and uncertainty values which are also bounded in $[0,1]$. The Bayesian probability update from Equation (3.4) is now redefined to accommodate this probabilistic framework:

$$p(c \mid x_{1:t}, z_{1:t}) \;=\; \frac{p(z_t \mid c, x_t)\; p(c \mid x_{1:t}, z_{1:t-1})}{p(z_t \mid x_{1:t}, z_{1:t-1})\;}. \tag{4.9}$$

where c is a single grid cell, $x_{1:t}$ is the set of poses of the robot till time t and $z_{1:t}$ is the set of observations of the robot till time t. The observations of the robot in our approach are RGB images I_t^{rgb} The term $p(z_t \mid m, x_t)$ represents the likelihood obtained directly from the uncertainty estimation process. In our approach, the likelihood is assumed to be a Gaussian distribution parametrized by the mean and variance derived from the combination of context and viewpoint uncertainties, defined as $\hat{\mu_S}$ in Equation (4.4) and $\hat{\sigma_S}^2$ in Equation (4.8) respectively. The term $p(m_i \mid x_{1:t}, z_{1:t-1})$ represents the prior probability of the grid cell, which we assume to also be a Gaussian distribution parametrized by the mean $\mu_{c,0}$ and variance $\sigma_{c,0}^2$ at t=0. Since both the prior and likelihood are Gaussian, the posterior distribution will also be Gaussian due to the conjugate prior property of Gaussian distributions. The map update can thus be formulated as a fusion of two Gaussians. This results in simple and computationally efficient probabilistic updates, as the fusion of Gaussian distributions involves straightforward analytical expressions for updating the mean and variance, making the process less resource-intensive compared to non-Gaussian updates.

We now define the measurement at time t as $z_t = [\hat{\mu_{S,t}}, \hat{\sigma^2_{S,t}}]$, such that $\hat{\mu_{S,t}}$ is the mean, and $\hat{\sigma^2_{S,t}}$ is the variance of semantic relevance at time t respectively. Both $\hat{\mu_{S,t}}$ and $\hat{\sigma^2_{S,t}}$ are derived from the sensor model we developed in Section 4.2 and are assumed to be drawn from a Gaussian distribution. Considering the Gaussian prior distribution over each grid cell an the likelihood, we adapt the formulation of the Bayesian map update defined in Section 3.3 to compute the posterior. The posterior of each grid cell on the map at time t m_t, as parametrized by the posterior mean and variance $\mu_{c,t}$ and $\sigma_{c,t}^2$ can be updated as:

$$\mu_{c,t} = \frac{\sigma_{c,t-1}^2\; \mu_{S,t} + \sigma_{S,t}^2\; \mu_{c,t-1}}{\sigma_{c,t-1}^2 + \sigma_{S,t}^2}\;, \tag{4.10}$$

$$\sigma_{c,t}^2 = \frac{\sigma_{c,t-1}^2 \, \sigma_{S,t}^2}{\sigma_{c,t-1}^2 + \sigma_{S,t}^2} \, . \tag{4.11}$$

This probabilistic formulation enables us to update the semantic relevance of regions in the environment in our metric-semantic context map. In our implementation, we fix the value of the prior mean $\mu_{c,0}$ as 0.5 and the prior variance $\sigma_{c,0}^2$ as 0.5 at t=0. The choice of 0.5 as the initial mean reflects that the cell is considered neither semantically relevant nor entirely irrelevant to the target based on available information. Similarly, setting the initial variance to 0.5 indicates that the initial belief is neither entirely uncertain nor completely certain. In the next section, we detail on how our context map can be used for downstream informative planning for exploration of semantically relevant regions.

4.4 Uncertainty-Informed Exploration

Building on the probabilistic map representation introduced in the previous section, we now transition to uncertainty-informed exploration strategies that utilize the map to guide the robot's actions. The context map, which integrates both context and viewpoint uncertainties, provides a continuously updated estimate of the semantic relevance and uncertainty associated with different regions in the environment. This rich information allows for more effective planning and decision-making by enabling the robot to prioritize areas with high potential for discovering the target object while accounting for uncertainty.

Uncertainty-informed exploration strategies aim to balance the trade-off between selecting regions with high estimated value based on the current map (exploitation) and exploring regions with greater uncertainty to gain additional information (exploration) [81, 100, 111, 99]. By leveraging the probabilistic framework, these strategies dynamically adjust the robot's actions in response to evolving conditions and new observations. This section introduces two uncertainty-informed planning approaches designed to enhance the robot's efficiency and effectiveness in object search tasks. The first approach is uncertainty-informed frontier exploration, a myopic strategy that performs single-step utility maximization by weighing the mean and uncertainty of map frontiers, using reward functions to select the most promising frontiers for further investigation, as outlined in Section 4.4.1. The second approach is uncertainty-informed monte carlo tree search (MCTS), a non-myopic strategy that formulates the navigation task as a MDP and utilizes MCTS make sequential decisions over multiple future actions by evaluating the semantic relevance and uncertainty of grid cells on the map, to optimize long-term gains. This approach is described in Section 4.4.2

4.4.1 Uncertainty-Informed Frontier Exploration

In uncertainty-informed frontier-based exploration, we leverage our maps to navigate to frontiers, which are regions on the periphery of explored and unexplored space. Frontiers are derived from the occupancy grid map, represented by the 2D position of the grid cell at the center of each frontier in the obstacle map M_O:

Let $\mathbf{x}_{f_i} = (x_{f_i}, y_{f_i})$ represent the 2D position of the grid cell at the center of frontier f_i, where $i = 1, 2, \ldots, N$ denotes each individual frontier. The probability of contextual relevance of each frontier f_i is characterized by the expected semantic relevance μ_{f_i} and variance $\sigma^2_{f_i}$ from the context map, as:

$$\mu_{f_i} = \mu(\mathbf{x}_{f_i}) \ , \tag{4.12}$$

$$\sigma^2_{f_i} = \sigma^2(\mathbf{x}_{f_i}) \ . \tag{4.13}$$

Here, $\mu(\mathbf{x}_{f_i})$ and $\sigma^2(\mathbf{x}_{f_i})$ represent the expected contextual relevance and the associated uncertainty (variance) for each frontier position $\mathbf{x}_{f_i}$. This probabilistic representation captures the uncertainty associated with the contextual relevance of each frontier with respect to the target object. To guide the robot's exploration, we use two reward functions: expected improvement (EI) [48] and gaussian process upper confidence bound (GP-UCB) [108], applied as exploration policies I-FBE1 as detailed in Section 4.4.1.1 and I-FBE2 as explained in Section 4.4.1.2, respectively. By considering both contextual relevance and uncertainty, these policies enable a more informed search for the target object.

4.4.1.1 I-FBE1

In frontier exploration, the objective is to select frontiers on a map that offer the highest potential for finding the target object. This task can be framed within the context of Bayesian optimization, where the probabilistic context map serves as the surrogate model which quantifies the uncertainty of contextual relevance of regions on the map. In Bayesian optimization, acquisition functions are used to approximate the true black-box function, in this case, the contextual relevance across regions on the map.

By estimating the potential improvement over the current best observation, we maximize an acquisition function. Each frontier pose $\mathbf{x}_{f_i}$ is treated as a candidate to explore, and we use the expected improvement acquisition function to evaluate the potential reward of visiting each frontier. The expected improvement [48] measures the expected improvement over the current highest observed mean μ_{best} among the frontiers on the map, guiding the robot towards frontiers that are likely to provide the most informative observations. Given N frontiers at a timestep t, the reward for a frontier using expected improvement can be expressed as:

$$\mathrm{EI}(f) = (\mu_f - \mu_{\text{best}}) \cdot \Phi\left(\frac{\mu_f - \mu_{\text{best}}}{\sigma_f}\right) + \sigma_f \cdot \phi\left(\frac{\mu_f - \mu_{\text{best}}}{\sigma_f}\right) , \tag{4.14}$$

where μ_{best} is the highest mean among the frontiers on the map at timestep t. Φ and ϕ are the cumulative distribution function and the probability density function of the standard normal distribution. The term $\frac{\mu_f - \mu_{\text{best}}}{\sigma_f}$ represents the improvement normalized by the uncertainty at the frontier. In the I-FBE1 exploration policy, the EI criterion is used as a reward function to prioritize frontiers that not only have a high estimated relevance but also hold significant uncertainty, offering a chance to discover regions with higher semantic relevance than currently known. This approach balances the exploration-exploitation trade-off by selecting frontiers based on the mean and uncertainty from the context map, as defined in Equation (4.12) and Equation (4.13) respectively. Additionally, exploring high uncertainty regions in turn, reduces the overall map uncertainty.

4.4.1.2 I-FBE2

While the expected improvement reward is a widely popular acquisition function for Bayesian optimization, its effectiveness to guide exploration by selecting frontiers is understudied. Additionally, since the function prefers high potential gains over the current best estimate, it is known to be greedy [84], i.e, it tends to focus on regions where the predicted mean similarity is relatively high. In the frontier exploration scenario, this can lead to under-exploration of areas with high uncertainty but lower immediate expected gains, potentially causing the robot to miss opportunities to discover more informative regions that are less certain but could offer significant insights if explored. To address this limitation, and model exploration of uncertain regions explicitly, we use the Gaussian Process Upper Confidence Bound (GP-UCB) criterion [108]. GP-UCT explicitly balances the trade-off between exploration and exploitation. The GP-UCB criterion is a well-known acquisition function in Bayesian optimization, designed to encourage exploration in uncertain regions while still considering predicted relevance. GP-UCB uses this probabilistic representation to guide the robot's exploration by combining the predicted mean and uncertainty to evaluate each frontier's overall utility. GP-UCB is defined as:

$$\text{GP-UCB}(f) = \mu_f + \sqrt{\beta}\sigma_f , \tag{4.15}$$

where β is a hyperparameter that controls the trade-off between exploration and exploitation. A higher β encourages exploration of more uncertain frontiers, while a lower β focuses on exploiting frontiers with higher mean. For our

approach, we use $\beta = 2$ as this value is suitable for balancing exploration and exploitation when the means and uncertainties are bounded between [0,1].

By employing GP-UCB in the I-FBE2 exploration policy, we enhance the robot's ability to explore uncertain areas that may otherwise be neglected by EI. This approach dynamically adjusts the exploration strategy, allowing the robot to discover new regions with potentially high semantic relevance while also exploiting areas with known high similarity. We evaluate the differences between these two frontier policies through extensive experiments in Section 5.2.2.

4.4.2 Uncertainty-Informed MCTS Exploration

The task of ObjectNav requires a robot to locate a user-defined target object in an unknown environment based on its observations. This task involves making decisions at each time step about where to move next, thereby characterizing it as a sequential decision-making problem. Such problems can be formulated as MDPs, as discussed in Section 3.4. This formulation allows for the application of various optimal policy search methods, including MCTS, which has demonstrated success in complex decision-making tasks such as strategy games, autonomous planning, and robot navigation [98, 107].

We formulate ObjectNav as a Markov Decision Process (MDP), defined by the tuple (S, A, T, R), where :

- $\mathcal{S}$: The set of states, which consists of the pose of the robot $\mathbf{x}_t^w$, the obstacle map, the visit map and the context map.
- $\mathcal{A}$: The set of actions, with an action selected from this set as $a \in \mathcal{A}$
- $T(s, a, s')$: The transition model, which defines the probability of arriving at state s' after taking action a in state s. In our formulation, we assume the transitions are unitary and deterministic.
- $R(s, a, s')$: The reward function, which provides feedback based on the robot's progress in locating the target object.

The objective in this MDP formulation is to find the optimal policy π, which maps states to actions that maximize the expected cumulative reward, guiding the robot to locate the target object efficiently. In subsequent subsections, we elaborate on our state space, reward formulation and action space design which then would enable the application of MCTS as defined in Section 3.4 as a planning strategy for ObjectNav.

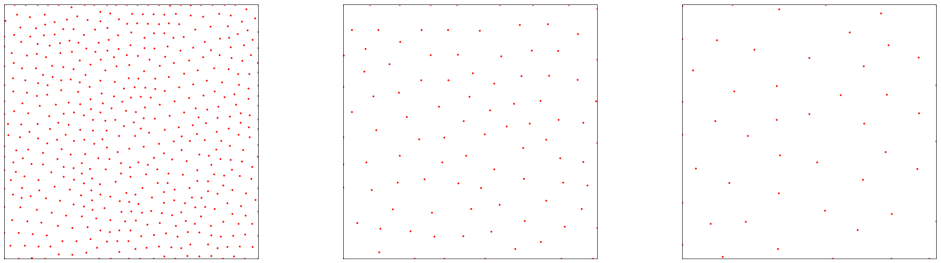


Figure 4.5: This image depicts sampling of points in a 2D grid using Farthest Point Sampling (FPS) [39] with (left) 500 samples, (center) 100 samples, (right) 50 samples. FPS samples equidistant points from the grid. We use FPS in our action space design to plan with a sparse set of states from our 2D grid map, reducing computational complexity.

4.4.2.1 Action Space Design

Defining an effective action space is crucial for the success of MCTS. In the context of ObjectNav, the action space determines the possible movements an agent can take while navigating its environment. A well-defined action space not only facilitates efficient exploration of the environment but also ensures that the agent can make informed decisions based on available options. In complex scenarios, where the agent must traverse large maps, an overly expansive action space can lead to inefficient exploration and increased computational overhead. This holds true for our grid map representations which represent complex household environments with grid cells.

In order to sample actions from the dense map representation, we propose an action space utilizing the Farthest Point Sampling (FPS) [83] algorithm, specifically through an optimized method called bucket-based Farthest Point Sampling (BFPS) [39]. BFPS samples a uniform representative set of points from the input point set by finding the subset of points that are the farthest away from each other. This approach organizes a large-scale set of points into a two-level tree data structure with multiple buckets. BFPS operates in two main stages: first, it constructs a KD-tree to organize the points into these buckets. Then, during the sampling stage, it selectively processes only the necessary buckets, which reduces memory usage and speeds up the action selection. For instance, it filters out buckets that aren't relevant to the current sampling, focusing only on those that contribute to the maximum distance from reference points. BFPS has state-of-the-art runtime speeds among FPS algorithms, thus justifying its use in real time exploration tasks such as ObjectNav.

We utilize FPS in our action space design to sample representative positions of states from the entire context map M_C. We sample ω potential actions by sampling ω samples from M_C using BFPS, effectively reducing the action space

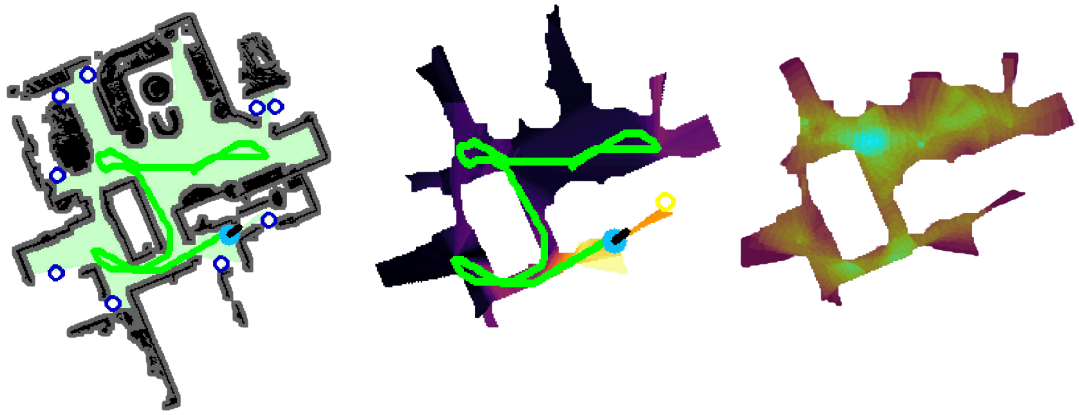


Figure 4.6: Maps used in our approach: Obstacle map (left) - where obstacles are in black, clearance distance from obstacles are in gray, explored area is in green and frontiers are denoted in dark blue circles. The robot is denoted as a light blue circle and its trajectory in high contrast green color. Context map (center) - denotes the map belief, where darker regions are lower valued and brighter regions are higher valued, thus more promising to explore. View Map (right) denotes the areas that the robot has already seen, where brighter regions have been more frequently seen.

size by selecting representative positions throughout the map. Formally, the action space $\mathcal{A}$ comprises of ω equidistantly sampled 2D positions on M_C, where ω is a hyperparameter. An action $a \in \mathcal{A}$ will be a 2D position v_a^w. To give an intuition to the reader, an example of the FPS algorithm on different sample sizes is illustrated in Figure 4.5.

4.4.2.2 State Space and Reward Design

In our formulation of ObjectNav, the state space $\mathcal{S}$ encapsulates the essential information about the robot and the environment required for planning. Each state $s \in S$ represents the robot's current understanding of the environment, comprising the robot pose $\mathbf{x}_t^w$ at each timestep, the obstacle map M_O, context map M_C and the view map M_V. On the basis of the state space and the action space previously defined in Section 4.4.2.1, we can formulate the reward function $R(s, a, s')$ for our MCTS-based exploration strategy. The action space $\mathcal{A}$ has $\gamma = |\mathcal{A}|$ actions. The evaluation of the reward of each action is done via our reward function, which comprises of three components, the semantic relevance reward $R_S(v_a^w, M_C)$, which is computed on using the position of the action and the context map, the exploration reward $R_e(v_a^w, M_O)$, computed on the basis of

the position of the action and obstacle map and the revisit penalty $R_r(v_a^w, M_V)$, dependent on the action position and the view map.

The context map M_C is a key element in the state space, representing the semantic relevance and uncertainty associated with different regions in the environment, as discussed in Section 4.3. By incorporating the context map into the state space, the planning process can focus on areas with high potential semantic relevance, this results in the reward for semantic relevance as:

$$R_S(v_a^w, M_C) = M_C[v_a^w] \ . \tag{4.16}$$

The robot's pose provides the position and orientation of the robot in the environment, which is essential for collision-free trajectory planning. The obstacle map M_O indicates occupied and free regions. In addition, providing information about the frontiers, which represent areas that if explored, can lead to an expansion in the current map. The exploration reward is modeled as a reward that encourages the robot to explore actions near the frontiers from the obstacle map M_O, by minimizing the distance to closest frontier v_f^w from v_a^w to expand the map, defined as:

$$R_e(v_a^w, F) = \frac{1}{1 + min\ d(v_f^w, v_a^w)} \ . \tag{4.17}$$

In addition to spatial and semantic information, the state space includes a view map M_V that records the frequency with which different areas have been explored. This component is crucial for prioritizing unseen or less frequently seen regions, thereby reducing redundant exploration and increasing the efficiency of the search process. By tracking exploration history, the view map ensures that the planning algorithm directs the robot toward areas that are more likely to yield new information, reducing the occurrence of re-views of a region. The re-view penalty R_V is defined based on the count of the grid cell from the view map at the current robot's pixel location:

$$R_r(v_a^w, M_V) = \frac{1}{e^{M_V v_a^w}} \quad | \quad e^{M_V v_a^w} > 0 \ , \tag{4.18}$$

where reward has an upper bound of 1 exponentially decreases the reward as the frequency of the gridcell on M_V increases. The combined reward function is the weighted sum of the cosine reward and exploration bonus, scaled by the penalty factor, is expressed as:

$$R \ = \ \alpha \cdot R_S(v_a^w, M_C) + (1 - \alpha) \cdot (R_e(v_a^w, F) \cdot R_r(v_a^w, M_V)) \ , \tag{4.19}$$

where α denotes a hyperparameter that balances between the exploitation of high semantic relevance values on with the exploration of novel areas by traversing near frontiers, which may result in a extension of the map. This comprehensive state and reward formulation enables the MCTS-based exploration strategy to dynamically update as new observations are gathered, ensuring that the planning algorithm operates with the most current information. The reward facilitates informed decision-making, allowing the robot to navigate complex environments and effectively search for the target object. By accounting for physical constraints, semantic relevance and redundant exploration, the state space and reward design enhances the robot's ability to perform the ObjectNav task in a structured and adaptive manner.

Chapter 5

Experiments

5.1 Experimental Setup

Our approach focuses on locating and navigating to objects in indoor environments, particularly households. To evaluate its effectiveness, we test it in realistic 3D household environments using the high-fidelity Habitat simulator [101]. Habitat is a flexible, high-fidelity and high-performance 3D simulator written in C++ that offers a user-friendly Python API, supporting user-configurable robots, sensors, and a variety of 3D datasets. The simulator facilitates high fidelity and accurate environment simulations by allowing simulation on mesh representations of 3D environments including house structures and objects. Habitat has been the standard platform for the ObjectNav Challenges [6, 123, 121], which serve as competitions to evaluate and benchmark different approaches to solve the task. Subsequently, it has become the de facto tool for evaluating ObjectNav solutions, widely adopted in the research community [113, 82, 17, 6, 44, 128].

Figure 5.1: Habitat [101] is a widely adopted high-fidelity simulator that supports configurable robots, sensors, and a wide variety of 3D datasets. These features make it an ideal platform for evaluating our approach.

5.1.1 Datasets

Figure 5.2: Our ObjectNav approach is evaluated in the Habitat simulator [101] on the HM3D [86] and MP3D [16] datasets, which consist of 3D scans of multi-floor household environments. The image shows two example environments from the HM3D dataset: a single-floor house (left) and a multi-floor house (right). Courtesy: *AI Habitat*

Evaluating open-vocabulary ObjectNav pipelines is challenging because existing datasets often lack comprehensive annotations for a wide range of objects and environments. To address this, our evaluation setup follows the criteria used in prior ObjectNav work [128] which evaluate their approach on multiple datasets. We evaluate our approach on the Matterport3D (MP3D) [16] and Habitat-Matterport 3D (HM3D) [86] datasets. These datasets offer real-world reconstructions of complex multi-floor indoor environments such as homes, offices, and commercial areas with diverse layouts and objects, making them suitable for evaluating ObjectNav pipelines and have been widely used by the community [17, 72, 30].

The MP3D dataset is a foundational yet challenging resource, featuring 3D mesh reconstructions of 90 buildings created from RGB-D scans. It includes rich semantic annotations, supporting tasks such as object recognition and scene segmentation. However, MP3D's limited number of unique environments can restrict the evaluation of generalization, and its 3D scans often suffer from occlusions, incomplete data, and noise in cluttered or complex spaces, requiring approaches to handle partial visibility and gaps. These imperfections make MP3D a challenging testbed for navigation and scene understanding. Building on the foundation laid by MP3D, the HM3D dataset significantly expands the scope and diversity of 3D environments. HM3D on the other hand comprises over 1,000 high-quality reconstructions of indoor spaces, offering a much larger and more varied set of environments for training and evaluating approaches. Its greater scale addresses the limitations of MP3D, providing diverse room layouts, object configurations, and lighting conditions that are more representative of real-world variability. HM3D is particularly valuable for testing generalization capabilities, as it includes more complex and cluttered environments that closely mimic real-world challenges.

Dataset	Total Environments	Total Objects	Total Episodes
HM3D	20	6	2000
MP3D	11	21	2195

Table 5.1: Parameters of the HM3D and MP3D datasets for ObjectNav evaluation.

In this thesis, we evaluate our ObjectNav approach using the Habitat simulator on the validation splits of both MP3D and HM3D datasets. This split consists of complex household environments suitable for thorough evaluation. An overview of the comparison between the two datasets is provided in Table 5.1. The evaluation spans 31 unique house configurations and a fixed number of episodes per dataset, with each episode having 500 timesteps. HM3D, with $2,000$ episodes across 20 environments, offers a larger variety of unique environments but is limited to 6 object categories from the MS-COCO dataset. Conversely, MP3D features $2,195$ episodes across 11 environments, covering 21 object categories, out of which 15 objects are non-COCO objects, making it a nuanced testbed due to its broader object diversity and visual imperfections. Together, these datasets enable a comprehensive evaluation of our ObjectNav pipeline and facilitate a balanced assessment of its performance across varied environments and objects.

5.1.2 Evaluation Metrics

In the ObjectNav problem, several metrics are used to evaluate the performance of agents by the research community [1, 6, 112] including "success rate" (SR), "success weighted by path length" (SPL), "Soft SPL", and "distance to goal" (DTG). Each metric captures different aspects of the agent's navigation behavior, from basic task completion to the efficiency and quality of the path taken. The SR is the simplest metric and measures whether the agent successfully reaches the target object. An episode is considered successful if the agent stops within a predefined distance of 1 meter from the target object and performs a a_{stop} action. The success rate is calculated as the ratio of successful episodes to the total number of episodes. While this metric evaluates basic task completion, it does not account for the efficiency of the path taken or whether the agent's path was optimal. To account for path efficiency, the SPL metric is used. SPL compares the length of the actual path taken by the agent with the shortest possible path to the goal, which is provided as ground truth. It is defined as:

$$\text{SPL}_i \quad = \quad S_i \cdot \frac{l_i}{\max(p_i, l_i)} \ , \tag{5.1}$$

where:

- S_i is the success indicator for the i-th episode, where $S_i = 1$ if the agent succeeds and $S_i = 0$ otherwise.
- l_i is the length of the shortest path to the goal.
- p_i is the length of the actual path taken by the agent.

The overall SPL score is the average over all episodes:

$$\text{SPL} = \frac{1}{N} \sum_{i=1}^{N} \text{SPL}_i \ . \tag{5.2}$$

SPL ensures that agents are not only judged on whether they reach the goal but also on how efficiently they do so, penalizing unnecessary deviations from the ground truth path. However, SPL treats all failures equally, offering no distinction between agents that nearly reached the goal and those that failed to make substantial progress. To address this limitation, Soft SPL introduces partial credit for agents that make significant progress toward the goal but do not fully reach it. Unlike standard SPL, which assigns a success score of 0 for any failure, soft SPL rewards agents based on their proximity to the target, providing a more nuanced assessment of performance, particularly for near-successful attempts. Lastly, the DTG metric measures how far the agent is from the target when it stops. This metric is typically represented as the Euclidean distance between the agent's final position and the target object. It is especially useful for distinguishing between partial successes and complete failures, as it gives an indication of how close the agent came to reaching the target, thereby complementing the SR, SPL, and Soft SPL metrics. We use all four of these metrics in our evaluation, however most state-of-the-art approaches only compare with two metrics, SR and SPL, which are directly related to success. Therefore when comparing our work to the state-of-the-art ObjectNav approaches, as shown in Section 5.2.4, we focus solely on these two metrics.

5.1.3 Baselines

To evaluate our proposed methods, we compare them against three baseline methods: a closest frontier policy, a random frontier policy, and VLFM [128], a state-of-the-art approach for ObjectNav. The closest frontier policy directs the robot to the closest frontier during exploration. The random frontier policy selects a random frontier for the agent to explore at each timestep. VLFM is a semantically informed frontier policy, leveraging cosine similarities from VLMs to select the most suitable frontier for exploration. This approach is the most similar to ours. It consistently outperforms other frontier-based exploration methods [30, 132, 72],

making it a strong baseline for comparison in this study. In addition, we also evaluate against other recent methods which tackle this problem.

5.1.4 Parameters

The MCTS planner used in this work is configured with a set of parameters optimized for effective exploration. Table 5.2 outlines the key parameters of our MCTS policy. These parameters have been chosen to balance between performance and computational complexity. Iterations are needed per decision, with a search depth of 3, allowing it to consider up to three future action steps. The discount factor is set to 1, indicating that future rewards are not discounted, giving equal importance to all potential future outcomes. The UCT exploration constant is set to $1/\sqrt{(2)}$, ensuring a balanced trade-off between exploration and exploitation during node selection, thus allowing the planner to explore less-visited nodes while also prioritizing high-reward actions.

Parameter	**Value**
Iterations	50
Depth	3
Discount Factor	1
UCT Exploration Constant	0.707

Table 5.2: Parameters for MCTS

Table 5.3 lists the parameters related to the action space of the MCTS planner. The planner uses farthest point sampling (FPS) to select 100 sample points per decision, enhancing the diversity of candidate actions by choosing points that maximize the distance between them. Each node in the search tree can choose from 3 possible actions, allowing the planner to explore multiple potential action sequences at each step. These parameters ensure that the MCTS planner effectively navigates complex environments by maintaining a robust exploration strategy while being computationally feasible for real-time applications.

Parameter	Value
FPS Samples	100
Action per Node	3
Reward Hyperparam α	0.5

Table 5.3: Parameters for MCTS action space design

5.2 Performance Evaluation

5.2.1 Effect of data uncertainty on VLFM

In the experiment shown in Table 5.4, we investigate how variations in prompts affect the performance of our baseline, VLFM. We assess the impact of data uncertainty on the ObjectNav problem by varying text prompts. By changing the prompts used to describe the same target object, we analyze how different phrasings influence the pipeline's performance at the ObjectNav task. This analysis aims to understand the sensitivity of the VLFM approach to data uncertainty, providing insights into optimizing prompt design and improving robustness in the pipeline's real-world deployment.

Our analysis reveals that modifying the prompt to "Seems like a `target_object` is ahead" a seemingly minor adjustment that removed the word "there", resulted in a 0.6% increase in the success rate of object detection. Conversely, using only the name of the object as the prompt resulted in a nearly 2% decrease in the success rate. These findings indicate that downstream robotics tasks are sensitive to the inherent uncertainty in VLMs' contextual encoding, where variations in prompt choice can significantly impact performance. This sensitivity reveals a form of data uncertainty propagation, as changes in prompt formulation affect how context is represented and utilized in decision-making processes.

Prompt	**SR ↑**	**SPL ↑**	**Soft SPL ↑**	**DTG ↓**
Seems like there is a target_object ahead	52.6%	30.42	36.3	4.1301
A place where target_object can be found	51%	29.71	36.43	4.1085
A target_object can be in the vicinity	53.20%	31.20	36.82	4.2236
Seems like a target_object is ahead	53.20%	30.50	36.17	4.1391
A target_object is in the vicinity	51.65%	28.67	34.32	4.2399
target_object likely ahead	52.45%	29.86	35.84	4.1848
target_object	50.60%	28.28	34.76	4.1962

Table 5.4: Performance metrics for different prompts on the HM3D dataset.

5.2.2 Informative Frontier Exploration

We compare the performance of the VLFM approach with traditional frontier-based exploration methods on the HM3D validation set in Table 5.5. Frontier-based exploration typically guides the robot towards the boundaries between explored and unexplored regions, aiming to maximize coverage of the environment. In our evaluation, we compare against three frontier exploration variants: Greedy-FBE, which navigates to the closest frontier from the robot's position; Random-FBE, which selects a random frontier; and VLFM, which prioritizes

Planner	**SR ↑**	**SPL ↑**	**Soft SPL ↑**	**DTG ↓**
Closest-FBE	11.80	9.34	20.26	6.5845
Random-FBE	37.30	23.32	32.38	4.8057
VLFM	52.60	30.42	36.30	4.1301
I-FBE1 (Prompt Uncertainty)	51.95	27.07	32.41	4.2418
I-FBE1 (Prompt + Viewpoint Uncertainty)	52.25	28.96	34.66	4.2615
I-FBE2 (Prompt + Viewpoint Uncertainty)	52.60	26.81	32.23	4.2021

Table 5.5: Performance metrics of different frontier based exploration planners on the HM3D validation set. We compare Closest-FBE, a frontier exploration baseline that navigates to the closest frontier; Random-FBE, a frontier based exploration baseline that navigates to a random frontier and VLFM, a frontier exploration baseline which navigates to the frontier with highest cosine similarity; with our informative frontier-based exploration approaches.

the frontier with the highest cosine similarity score derived from a VLM. While traditional frontier-based methods rely solely on spatial exploration, the VLFM approach integrates semantic information to guide exploration towards regions that are more likely to contain the target object and is the closest to our approach.

In our analysis we find that the Closest-FBE policy performs the worst due to the fact that by going to the nearest frontier, the robot might get stuck in exploring narrow close-by regions which can be located away from the target object. The Random-FBE policy performs better because by choosing random frontiers, it does not get stuck in narrow search space and can cover more area, although still not being goal-directed. Then as a strong baseline, we compare against the state-of-the-art ObjectNav baseline, VLFM. Next, we compare our approach I-FBE1 with two different kinds of uncertainties. First, using only the uncertainty derived from prompt ensembles and then using the combination of prompt ensemble and viewpoint uncertainties. We observe that I-FBE1 using both prompt and viewpoint uncertainty performs better in all metrics compared to using only prompt uncertainty. Finally we compare with the IFBE-2 policy, which has a success rate equal as the state-of-the-art baseline but has a lower SPL than IFBE or VLFM. This is due to the fact that it explores much more. By comparing these strategies, we aim to evaluate the benefits of incorporating semantic guidance through VLMs versus relying purely on spatial coverage. This comparison provides insights into how effectively each approach can navigate cluttered and dynamic environments, with a focus on finding objects more efficiently and improving the overall performance of the robot's exploration and navigation tasks.

Action Space	SR ↑	SPL ↑	Soft SPL ↑	DTG ↓
Low Level Action Space	14.00	7.88	13.41	8.04
Radius Based Sampling	28.50	12.38	15.26	5.9178
Farthest Point Sampling (FPS)	36.50	**15.25**	**18.81**	5.3044
I-MCTS (FPS + Expected Improvement Reward)	**36.90**	15.08	18.45	**5.2630**

Table 5.6: Performance of MCTS with different action spaces on the HM3D dataset.

5.2.3 Action Space design of MCTS

In the ablation study on MCTS action spaces, we compare the performance of three strategies: low-level action space, radius-based sampling, and Farthest Point Sampling (FPS), keeping the MCTS parameters unchanged. The results show that the low-level action space performs the worst, struggling with navigation due to its restricted action choices, which often lead to the robot getting stuck. Radius-based sampling provides better performance by offering improved navigation efficiency, but its limited action range reduces its effectiveness in more complex scenarios. In addition, the computational cost of sampling from the entire map, makes this approach intractable for real time planning. FPS delivers a substantial performance boost by effectively balancing exploration and computational demands through the selection of equidistant points across the map. When integrated with the expected improvement reward function in I-MCTS, FPS shows a slight additional improvement, highlighting the value of uncertainty-informed planning for goal-oriented exploration, though there is a minor trade-off in path optimality due to a stronger focus on exploration.

Overall, the ablation study highlights that FPS provides the best trade-off between computational efficiency and navigation performance, outperforming the low-level and radius-based action spaces in terms of success rate and DTG.

5.2.4 Benchmarking ObjectNav Approaches

In the final experiment, as shown in Table 5.7, we compare our proposed methods with several state-of-the-art approaches for the ObjectNav task across both of our benchmark datasets. Our methods, which include I-MCTS and two variations of uncertainty-informed frontier-based exploration (I-FBE1 and I-FBE2), are assessed in terms of their ability to effectively navigate towards a target on the metrics of SR and SPL, which are consistent with the metrics reported by other methods. Notably, SoftSPL and DTG are not included, as they are less commonly reported in existing works and do not directly indicate task success.

The results of this experiment demonstrate that our methods perform competitively with the state-of-the-art approach VLFM. Although I-MCTS does not achieve state-of-the-art performance, it outperforms several earlier frontier-based

Approach	Year	HM3D		MP3D	
		SR↑	SPL↑	SR↑	SPL↑
ZSON [72]	2023	25.5	12.6	15.3	4.8
CoW [30]	2023	-	-	7.4	3.7
ESC [132]	2023	39.2	22.3	28.7	14.2
VLFM [128]	2024	**52.6**	**30.4**	**36.4**	**17.5**
Ours (I-MCTS)	2024	36.90	15.08	26.92	11.40
Ours (I-FBE1)	2024	52.25	28.96	35.26	16.47
Ours (I-FBE2)	2024	**52.60**	26.81	35.81	16.63

Table 5.7: Comparison of state-of-the-art for ObjectNav on HM3D and MP3D datasets.

methods, suggesting that employing a distinct planning strategy such as MCTS shows promise for the ObjectNav task. The uncertainty-informed frontier exploration strategies, I-FBE1 and I-FBE2, deliver results that are closely aligned with VLFM, demonstrating similar levels of navigation effectiveness across datasets. These approaches effectively utilize uncertainty information to guide exploration, yielding outcomes that rival the performance of VLFM, which does not incorporate uncertainty into its planning framework. Our experiments highlight that the inclusion of uncertainty-informed strategies allows our methods to achieve state-of-the-art performance. While there are slight differences in navigation efficiency across our different approaches, they consistently demonstrate the value of leveraging uncertainty for informed exploration, confirming the effectiveness of our techniques in comparison with other advanced navigation algorithms.

5.2.5 Analysis of Failure Modes

To systematically assess the limitations of our approach, we categorize and quantify specific failure modes that impact our success on the ObjectNav task. We analyze the failure modes of our approaches, I-FBE2 and I-MCTS, and compare them with the failure modes of VLFM on the HM3D dataset. This analysis is visually summarized in Figure 5.4. We quantify failure modes into three primary categories: *detection failures*, *exploration failures*, and *navigation failures*.

Detection failures consist of two main types: *false positives*, where the object detection module mistakenly identifies a non-target object as the target, often due to dataset constraints that mark only specific instances of an object as valid; and *false negatives*, where the object detection module fails to detect the target even when it was observed. These errors can lead to premature episode termination if the robot either misidentifies or misses the target. *Exploration failures* arise from the robot's challenges in navigating complex, multi-floor environments and

include several types. *Wrong floor* navigation occurs when the robot mistakenly moves to a different floor than the one with the target object, due to limited vertical spatial awareness. *Unattempted floor change* happens when the target object is initialized on a different floor than the robot, but the robot fails to initiate a floor change, restricting its search to an incorrect floor. *Unnecessary floor change* involves the robot moving to another floor despite starting on the one where the target object was located, leading to inefficient exploration and reduced time on the correct floor. Finally, *insufficient exploration* occurs when both the robot and target are on the same floor, but the robot does not search thoroughly enough within the allotted time to locate the target. *Navigation failures* occur when the robot's navigation policy fails to guide it to the target object after detection. This can happen if the robot is unable to approach the target closely enough to meet the distance criterion, which we characterize as a *bad stop*. Another reason for a navigation failure is an *episode timeout* where the robot fails to reach the target before the episode ends, often due to late discovery of the target object or an inefficient navigation path.

VLFM's performance on the HM3D dataset shows a failure rate of approximately 47.4% across 2000 episodes. Detection failures account for about 50% of total failures and exploration failures make up 46.9% of total failures. Navigation failures are far less common, accounting for only 2.2% of total failures.

Our myopic exploration approach, I-FBE2 rivals VLFM on success rate and similarly has a 47.4% failure rate on HM3D. In this approach, Detection failures account for approximately 51.6% of total failures, primarily due to false positives. Another significant source of failure stems from exploration issues account for 46.9% of total failures, due to navigation to wrong floors by failing to travel stairs and attempt necessary floor changes, or by accidentally traveling stairs and performing unnecessary floor transitions. Navigation failures are minimal for our approach as compared to other failure types and comparable to VLFM, as both methods use DD-PPO [118] for low level navigation.

In contrast to myopic methods, our non-myopic policy I-MCTS demonstrates a 63.1% failure rate. Of these failures, 41.9% stem from false positives, while 57.4% result from exploration challenges, with insufficient exploration being the primary contributor. We attribute this limitation in exploration to the MCTS planner's lack of an explicit mechanism to expand observed space actively, as occurs in frontier-based approaches. This becomes particularly problematic in environments with narrow corridors, where the planner's inability to systematically explore constrained areas reduces search effectiveness. In I-MCTS, actions are sampled as equidistant points across the map, but without explicitly targeting frontiers, even though the reward function assigns value to proximity to these unexplored areas. This design leaves the exploration dependent on chance sam-

Figure 5.3: Qualitative analysis of I-MCTS performance in narrow corridor environments, highlighting common exploration failures due to limited map coverage. The examples illustrate instances where insufficient frontier exploration leads the robot to revisit previously observed areas, restricting successful target detection and navigation.

pling near frontiers, which is inconsistent since frontiers lie at the periphery of the map, where fewer actions are typically sampled. As a result, the observed map does not expand sufficiently, preventing full coverage of the environment. Consequently, most actions target open spaces rather than unexplored regions, resulting in a map that fails to expand sufficiently to cover the entire environment. This limitation causes the robot to repeatedly sample within known areas, revisiting the same locations despite penalization. Figure 5.3 illustrates several narrow corridor scenarios where this limitation is particularly evident.

Our failure mode analysis reveals that the two primary factors contributing to poor success rates in state-of-the-art ObjectNav approaches are detection-related and exploration-related failures. In Section 6.2, we discuss potential improvements to address these issues, focusing on enhancing detection accuracy and optimizing exploration strategies to improve performance in ObjectNav.

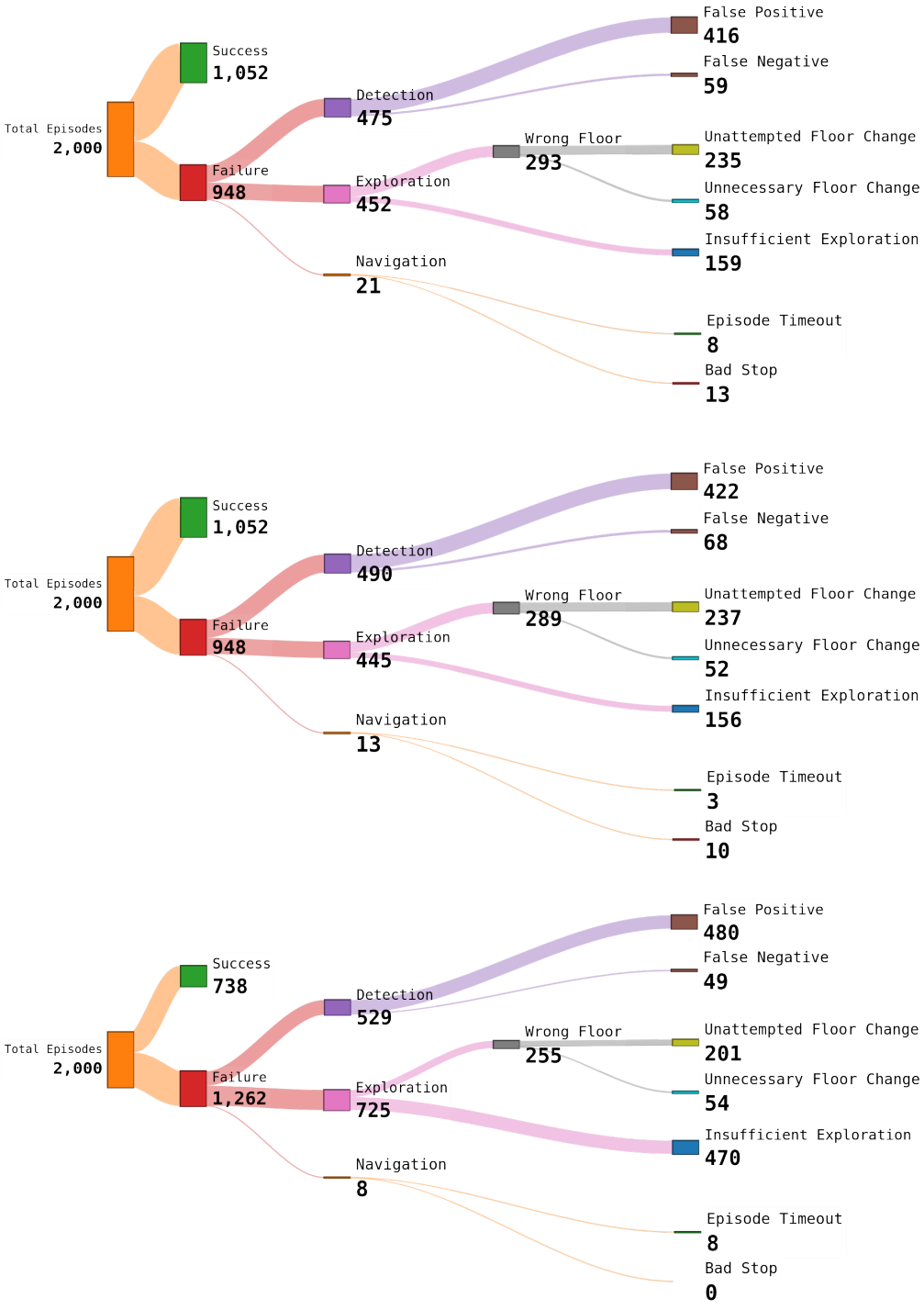


Figure 5.4: Failure mode comparison on the HM3D dataset: highlighting the performance differences between the state-of-the-art VLFM [128] approach (top) and our proposed methods, I-FBE2 (middle) and I-MCTS (bottom).

Chapter 6

Conclusion

In this thesis, we developed a novel training-free, open vocabulary semantic uncertainty-informed active perception pipeline for the ObjectNav problem. We implemented and evaluated our approach on different datasets to show the generalizability of our approach to different scenarios and objects. We then provided comparisons to other existing techniques. We provided a thorough experimental evaluation supporting all claims made in this thesis. Our results indicate that uncertainty informed active perception is a useful direction for semantic exploration in the ObjectNav task.

6.1 Short summary of key contributions

In conclusion, the key contributions of this ObjectNav pipeline proposed in this thesis are as follows:

- We used VLMs to semantically guide exploration towards semantically similar regions and detect objects.
- We demonstrated that VLMs are susceptible to data uncertainty when applied to downstream robotics tasks and proposed a method to quantify it.
- We developed a probabilistic map formulation to create an uncertainty-infused semantic map representation using VLM uncertainties.
- We developed information-theoretic rewards for myopic and non-myopic planners using our probabilistic map to make uncertainty-informed decisions.

6.2 Future Work

In this section, we discuss potential avenues for future work aimed to address the limitations identified in our approach. As outlined in Section 5.2.5, our quantitative analysis of failure modes, both for our method and the VLFM baseline, highlights two primary categories of issues: detection-related failures and exploration-related failures. Addressing these shortcomings will be essential for improving the overall performance and robustness of our approach.

Exploration-related failures make up a significant portion of the issues, especially when navigating stairs. The system often fails by not traveling stairs when needed or doing so unnecessarily. Additionally, insufficient exploration leads to poor coverage of the environment. To address these problems, future work should incorporate a hierarchical map representation, such as scene graphs, to retain spatial information across different levels. Currently, our approach resets the map after traveling stairs, causing a loss of previously explored areas. Hierarchical or graph-based maps could provide a more structured and efficient alternative to occupancy grids, improving tree-based planners like MCTS. This would allow planners to make better decisions about exploring different levels, such as when to travel stairs, and enable more sophisticated reward functions for efficient exploration and decision-making.

Detection-related failures, such as false positives and false negatives, pose significant challenges in our approach, worsened by the large scale of evaluation (about one million detection runs across 2000 episodes in the HM3D dataset). False negatives mainly arise from limitations within the perception system, leading to missed detections. In contrast, false positives are due to a combination of factors, inherent shortcomings in the perception system and ambiguities in the dataset. Specifically, the dataset may include multiple instances of the same object category, with only some considered valid for the task, causing the system to mistakenly end an episode when an incorrect instance is detected. Our current approach does not effectively separate perception limitations from dataset-related issues. Future work should prioritize developing advanced perception methods to address these complexities and improving datasets to minimize ambiguities, thereby enhancing detection accuracy and robustness.

Improving uncertainty quantification presents another significant opportunity for advancement. Currently, we address language uncertainty by randomly selecting five prompts in our ensemble, but adopting more diverse ensemble strategies or optimizing prompt selection could improve performance. Some techniques, such as CoOp [136] and CoCoOp [135], learn prompts, but these methods are tailored for tasks involving ImageNet-like classes, where objects are clearly present in images. There are currently no techniques that effectively associate objects with images where they are not visibly present, revealing a gap in perception

Image Augmentation	**SR ↑**	**SPL ↑**	**Soft SPL ↑**	**DTG ↓**
Original RGB Image	52.6%	30.42	36.3	4.1301
Horizontal Flip	51.90%	30.49	37.35	4.0336
Center Crop	50.30%	30.03	37.22	4.0715
Saturation(15%)	50.55%	30.14	36.94	4.1819

Table 6.1: Performance metrics for various image augmentations on the HM3D dataset, highlighting the impact of each augmentation on ObjectNav performance

Method	**SR ↑**	**SPL ↑**	**Soft SPL ↑**	**DTG ↓**
VLFM	52.6%	30.42	36.3	4.1301
VLFM + Object2Room	54.15%	32.63	38.87	3.9602

Table 6.2: Using the predicted room location instead of the object name improves all performance metrics.

that requires further exploration. Addressing other sources of data uncertainty, such as those introduced by image augmentations, is also essential. Our preliminary results show that applying augmentations before processing images through VLMs affects downstream tasks and performance, indicating the need to account for these factors in future uncertainty modeling efforts, as shown in Table 6.1. Incorporating such elements into a more comprehensive uncertainty quantification framework could lead to a more principled approach for active perception.

Beyond the current approach of directly associating objects with images, other relationships could be utilized, such as predicting the most likely room where an object might be found. This can be achieved by incorporating an LLM into the process to infer contextual relationships between objects and their typical locations. Our preliminary experiments using this approach showed promising results: mapping six objects from the HM3D dataset to their most likely rooms based on empirical observations led to a 1.5% improvement in success rate and a 2% improvement in SPL, as shown in Table 6.2 The mappings used were: “chair” to “living room,” “bed” to “bedroom,” “potted plant” to “living room/office,” “toilet” to “toilet/bathroom,” “tv” to “living room,” and “couch” to “living room.” These results suggest that incorporating spatial context into object navigation could enhance performance.

We believe that enhancing exploration strategies with flexible map representations, improving perception systems and datasets to reduce ambiguities, and quantifying other kinds of data uncertainties is needed to increase the robustness and efficiency of our approach and useful for the ObjectNav task.

6.3 Open source contributions

We plan to offer an open-source repository of our proposed ObjectNav pipeline, implemented in Python to ensure ease of use. A link to the repository is provided to support its evaluation and adoption by the robotics community.

- `https://gitlab.ipb.uni-bonn.de/utkarsh.bajpai/masterthesis`

Bibliography

[1] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R. Zamir. On Evaluation of Embodied Navigation Agents. *arXiv Preprint , arXiv:1807.06757*, 2018.

[2] R. Bajcsy. Active perception. *Proc. of the IEEE*, 76(8):966–1005, 1988.

[3] Ruzena Bajcsy, Yiannis Aloimonos, and John K. Tsotsos. Revisiting active perception. *Autonomous Robots*, 42(2):177–196, feb 2018.

[4] Ruzena Bajcsy and Mario Campos. Active and exploratory perception. *CVGIP: Image Underst.*, 56(1):31–40, jul 1992.

[5] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-Shot Object Detection. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, September 2018.

[6] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. ObjectNav Revisited: On Evaluation of Embodied Agents Navigating to Objects. *arXiv Preprint , arXiv:2006.13171*, 2020.

[7] Hriday Bavle, Jose Luis Sanchez-Lopez, Muhammad Shaheer, Javier Civera, and Holger Voos. S-graphs+: Real-time localization and mapping leveraging hierarchical representations. *IEEE Robotics and Automation Letters (RA-L)*, 2023.

[8] Richard Bellman. On a routing problem. *Quarterly of Applied Mathematics*, 16(1):87–90, 1958.

[9] I. Bogoslavskyi. *Robot Mapping and Navigation in Real-World Environments*. PhD thesis, Rheinische Friedrich-Wilhelms University of Bonn, 2018.

[10] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava

Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv Preprint , arXiv:2307.15818*, 2023.

[11] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners.

[12] Cameron B. Browne, Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A Survey of Monte Carlo Tree Search Methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43, 2012.

[13] Wolfram Burgard, Mark Moors, and Frank Schneider. Collaborative Exploration of Unknown Environments with Teams of Mobile Robots. In *Advances in Plan-Based Control of Robotic Agents*, pages 52–70, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[14] Tommaso Campari, Paolo Eccher, Luciano Serafini, and Lamberto Ballan. Exploiting Scene-Specific Features for Object Goal Navigation. In Adrien Bartoli and Andrea Fusiello, editors, *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, pages 406–421, Cham, 2020. Springer International Publishing.

[15] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, pages 9650–9660, October 2021.

[16] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.

[17] Devendra Singh Chaplot, Dhiraj Gandhi, Abhinav Gupta, and Ruslan Salakhutdinov. Object Goal Navigation using Goal-Oriented Semantic Exploration. In *Proc. of the Conf. Neural Information Processing Systems (NIPS)*, 2020.

[18] Boyuan Chen, Fei Xia, Brian Ichter, Kanishka Rao, Keerthana Gopalakrishnan, Michael S. Ryoo, Austin Stone, and Daniel Kappler. Open-vocabulary Queryable Scene Representations for Real World Planning. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 11509–11522, 2023.

[19] Junting Chen, Guohao Li, Suryansh Kumar, Bernard Ghanem, and Fisher Yu. How To Not Train Your Dragon: Training-free Embodied Object Goal Navigation with Semantic Frontiers. In *Proc. of Robotics: Science and Systems (RSS)*, 2023.

[20] Xuanming Cui, Alejandro Aparcedo, Young Kyun Jang, and Ser-Nam Lim. On the Robustness of Large Multimodal Models Against Image Adversarial Attacks . In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA, June 2024. IEEE Computer Society.

[21] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, Dec 1959.

[22] Vishnu Sashank Dorbala, James F. Mullen, and Dinesh Manocha. Can an Embodied Agent Find Your "Cat-shaped Mug"? LLM-Based Zero-Shot Object Navigation. *IEEE Robotics and Automation Letters (RA-L)*, 9(5):4083–4090, 2024.

[23] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. of the Intl. Conf. on Learning Representations (ICLR)*, 2021.

[24] Raphael Druon, Yusuke Yoshiyasu, Asako Kanezaki, and Alassane Watt. Visual Object Search by Learning Spatial Context. *IEEE Robotics and Automation Letters (RA-L)*, 5(2):1279–1286, 2020.

[25] Heming Du, Xin Yu, and Liang Zheng. Learning Object Relation Graph and Tentative Policy for Visual Navigation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, pages 19–34, Cham, 2020. Springer International Publishing.

[26] Joris Duguépéroux, Ahmad Mazyad, Fabien Teytaud, and Julien Dehos. Pruning playouts in monte-carlo tree search for the game of havannah. In Aske Plaat, Walter Kosters, and Jaap van den Herik, editors, *Computers and Games*, pages 47–57, Cham, 2016. Springer International Publishing.

[27] Kiana Ehsani, Tanmay Gupta, Rose Hendrix, Jordi Salvador, Luca Weihs, Kuo-Hao Zeng, Kunal Pratap Singh, Yejin Kim, Winson Han, Alvaro Herrasti, Ranjay Krishna, Dustin Schwenk, Eli VanderBilt, and Aniruddha Kembhavi. Imitating Shortest Paths in Simulation Enables Effective Navigation and Manipulation in the Real World. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[28] Karl Friston. The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11:127–138, 2010.

[29] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc' Aurelio Ranzato, and Tomas Mikolov. DeViSE: A Deep Visual-Semantic Embedding Model. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Proc. of the Conf. Neural Information Processing Systems (NIPS)*, volume 26. Curran Associates, Inc., 2013.

[30] Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 23171–23181. IEEE, 2023.

[31] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: Exceeding YOLO Series in 2021, 2021.

[32] Georgios Georgakis, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, and Kostas Daniilidis. Learning to Map for Active Semantic Goal Navigation. In *Proc. of the Intl. Conf. on Learning Representations (ICLR)*, 2022.

[33] Ross Girshick. Fast R-CNN. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[34] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014.

[35] Francesco Giuliari, Alberto Castellini, Riccardo Berra, Alessio Del Bue, Alessandro Farinelli, Marco Cristani, Francesco Setti, and Yiming Wang. POMP++: Pomcp-based Active Visual Search in unknown indoor environments. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1523–1530, 2021.

[36] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters. *IEEE Trans. on Robotics (TRO)*, 23(1):34–46, 2007.

[37] Qiao Gu, Alihusein Kuwajerwala, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Rama Chellappa, Chuang Gan, Celso M de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *Proc. of the Conf. on Robot Learning (CoRL)*, 2023.

[38] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021.

[39] Meng Han, Liang Wang, Limin Xiao, Hao Zhang, Chenhao Zhang, Xiangrong Xu, and Jianfeng Zhu. QuickFPS: Architecture and Algorithm Co-Design for Farthest Point Sampling in Large-Scale Point Clouds. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 42(11):4011–4024, 2023.

[40] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–524, 1988.

[41] P.E. Hart, N.J. Nilsson, and B. Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[42] Hermann von Helmholtz. Concerning the perceptions in general. *Treatise on physiological optics*, 3, 1867.

[43] A. Hornung, K.M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 34:189–206, 2013.

[44] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual Language Maps for Robot Navigation. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2023.

[45] N. Hughes, Y. Chang, and L. Carlone. Hydra: A real-time spatial perception system for 3D scene graph construction and optimization. In *Proc. of Robotics: Science and Systems (RSS)*, 2022.

[46] Lucas Janson and Marco Pavone. Fast marching trees: a fast marching sampling-based method for optimal motion planning in many dimensions-extended version. *Intl. Journal of Robotics Research (IJRR)*, pages 1–59, 2014.

[47] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. In Marina Meila and Tong Zhang, editors, *Proc. of the Intl. Conf. on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pages 4904–4916. PMLR, 18–24 Jul 2021.

[48] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[49] Abhishek Kadian, Joanne Truong, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra. Sim2Real Predictivity: Does Evaluation in Simulation Predict Real-World Performance? *IEEE Robotics and Automation Letters (RA-L)*, 5(4):6670–6677, 2020.

[50] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *Intl. Journal of Robotics Research (IJRR)*, 30(7):846–894, 2011.

[51] L.E. Kavraki, P. Svestka, J.-C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. on Robotics and Automation*, 12(4):566–580, 1996.

[52] Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but effective: Clip embeddings for embodied ai. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 14829–14838, 2022.

[53] Taewan Kim and Joydeep Ghosh. On Single Source Robustness in Deep Fusion Models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Proc. of the Conf. Neural Information Processing Systems (NIPS)*, volume 32. Curran Associates, Inc., 2019.

[54] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment Anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, October 2023.

[55] Levente Kocsis and Csaba Szepesvári. Bandit Based Monte-Carlo Planning. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Proc. of the Europ. Conf. on Machine Learning (ECML)*, pages 282–293, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[56] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, Aniruddha Kembhavi, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv preprint*, 2022.

[57] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Proc. of the Conf. Neural Information Processing Systems (NIPS)*, volume 25. Curran Associates, Inc., 2012.

[58] Rahul Kumar, Aditya Mandalika, Sanjiban Choudhury, and Siddhartha Srinivasa. LEGO: Leveraging Experience in Roadmap Generation for Sampling-Based Planning. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1488–1495, 2019.

[59] Mikko Lauri and Risto Ritala. Planning for robotic exploration based on forward simulation. *Journal on Robotics and Autonomous Systems (RAS)*, 83:15–31, 2016.

[60] S.M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[61] Chuyi Li, Lulu Li, Yifei Geng, Hongliang Jiang, Meng Cheng, Bo Zhang, Zaidan Ke, Xiaoming Xu, and Xiangxiang Chu. YOLOv6 v3.0: A Full-Scale Reloading, 2023.

[62] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large

Language Models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proc. of the Intl. Conf. on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pages 19730–19742. PMLR, 23–29 Jul 2023.

[63] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proc. of the Intl. Conf. on Machine Learning (ICML)*, pages 12888–12900. PMLR, 2022.

[64] Liunian Harold Li*, Pengchuan Zhang*, Haotian Zhang*, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded Language-Image Pre-training. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[65] Yimeng Li, Arnab Debnath, Gregory J. Stein, and Jana Košecká. Learning-Augmented Model-Based Planning for Visual Exploration. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 5165–5171, 2023.

[66] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, pages 740–755, Cham, 2014. Springer International Publishing.

[67] Dongfang Liu, Yiming Cui, Zhiwen Cao, and Yingjie Chen. Indoor Navigation for Mobile Agents: A Multimodal Vision Fusion Model. In *Intl. Joint Conf. on Neural Networks (IJCNN)*, pages 1–8, 2020.

[68] Peiqi Liu, Yaswanth Orru, Jay Vakil, Chris Paxton, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. OK-Robot: What Really Matters in Integrating Open-Knowledge Models for Robotics. *arXiv Preprint , arXiv:2401.12202*, 2024.

[69] 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.

[70] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Intl. Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004.

[71] Haokuan Luo, Albert Yue, Zhang-Wei Hong, and Pulkit Agrawal. Stubborn: A Strong Baseline for Indoor Object Navigation. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 3287–3293, 2022.

[72] Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. Zson: Zero-shot object-goal navigation using multimodal goal embeddings. In *Proc. of the Conf. Neural Information Processing Systems (NIPS)*, 2022.

[73] Oleksandr Maksymets, Vincent Cartillier, Aaron Gokaslan, Erik Wijmans, Wojciech Galuba, Stefan Lee, and Dhruv Batra. THDA: Treasure Hunt Data Augmentation for Semantic Navigation. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, pages 15354–15363, 2021.

[74] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proc. of the Intl. Conf. on Machine Learning (ICML)*, volume 48 of *Proceedings of Machine Learning Research*, pages 1928–1937, New York, New York, USA, 20–22 Jun 2016. PMLR.

[75] Amir Mobarhani, Shaghayegh Nazari, Amir H. Tamjidi, and Hamid D. Taghirad. Histogram based frontier exploration. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1128–1133, 2011.

[76] H. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, volume 2, 1985.

[77] H.P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, pages 61–74, 1988.

[78] OpenAI, Josh Achiam, Steven Adler, and 279 other authors not shown. GPT-4 Technical Report. *arXiv preprint*, 2024.

[79] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski.

DINOv2: Learning Robust Visual Features without Supervision. *arXiv Preprint , arXiv:2304.07193*, 2024.

[80] Giovanni Pezzulo, Thomas Parr, and Karl Friston. Active inference as a theory of sentient behavior. *Biological Psychology*, 186:108741, 2024.

[81] Marija Popović, Teresa Vidal-Calleja, Gregory Hitz, Jen Jen Chung, Inkyu Sa, Roland Siegwart, and Juan Nieto. An informative path planning framework for UAV-based terrain monitoring. *Autonomous Robots*, 44(6):889–911, jul 2020.

[82] Xavi Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Ruslan Partsey, Jimmy Yang, Ruta Desai, Alexander William Clegg, Michal Hlavac, Tiffany Min, Theo Gervet, Vladimír Vondruš, Vincent-Pierre Berges, John Turner, Oleksandr Maksymets, Zsolt Kira, Mrinal Kalakrishnan, Jitendra Malik, Devendra Singh Chaplot, Unnat Jain, Dhruv Batra, Akshara Rai, and Roozbeh Mottaghi. Habitat 3.0: A Co-Habitat for Humans, Avatars and Robots. *arXiv Preprint , arXiv:2310.13724*, 2023.

[83] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Proc. of the Conf. Neural Information Processing Systems (NIPS)*, volume 30. Curran Associates, Inc., 2017.

[84] Chao Qin, Diego Klabjan, and Daniel Russo. Improving the expected improvement algorithm. In *Proc. of the Conf. Neural Information Processing Systems (NIPS)*, NIPS'17, Red Hook, NY, USA, 2017. Curran Associates Inc.

[85] 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 *Proc. of the Intl. Conf. on Machine Learning (ICML)*, pages 8748–8763, 2021.

[86] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In *Proc. of the Conf. Neural Information Processing Systems (NIPS)*, 2021.

[87] J Redmon. You only look once: Unified, real-time object detection. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[88] Joseph Redmon and Ali Farhadi. YOLOv3: An Incremental Improvement, 2018.

[89] Dillon Reis, Jordan Kupec, Jacqueline Hong, and Ahmad Daoudi. Real-Time Flying Object Detection with YOLOv8. *arXiv Preprint , arXiv:2305.09972*, 2023.

[90] Allen Z. Ren, Jaden Clark, Anushri Dixit, Masha Itkina, Anirudha Majumdar, and Dorsa Sadigh. Explore until Confident: Efficient Exploration for Embodied Question Answering, 2024.

[91] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Proc. of the Conf. Neural Information Processing Systems (NIPS)*, volume 28. Curran Associates, Inc., 2015.

[92] Eduardo Romera, José M. Álvarez, Luis M. Bergasa, and Roberto Arroyo. ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation. *IEEE Trans. on Intelligent Transportation Systems (TITS)*, 19(1):263–272, 2018.

[93] A. Rosinol, A. Gupta, M. Abate, J. Shi, and L. Carlone. 3D Dynamic Scene Graphs: Actionable Spatial Perception with Places, Objects, and Humans. In *Proc. of Robotics: Science and Systems (RSS)*, 2020.

[94] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: an Open-Source Library for Real-Time Metric-Semantic Localization and Mapping. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2020.

[95] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, pages 2564–2571, 2011.

[96] Sohan Rudra, Saksham Goel, Anirban Santara, Claudio Gentile, Laurent Perron, Fei Xia, Vikas Sindhwani, Carolina Parada, and Gaurav Aggarwal. A Contextual Bandit Approach for Learning to Plan in Environments with Probabilistic Goal Configurations. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 5645–5652, 2023.

[97] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[98] Stuart Russell and Peter Norvig. *Artificial Intelligence, Global Edition A Modern Approach*. Pearson Deutschland, 2021.

[99] Julius Rückin, Liren Jin, Federico Magistri, Cyrill Stachniss, and Marija Popović. Informative Path Planning for Active Learning in Aerial Semantic Mapping. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2022.

[100] Julius Rückin, Federico Magistri, Cyrill Stachniss, and Marija Popović. An Informative Path Planning Framework for Active Learning in UAV-Based Semantic Mapping. *IEEE Trans. on Robotics (TRO)*, 39(6):4279–4296, 2023.

[101] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2019.

[102] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[103] Dhruv Shah, Michael Robert Equi, Blazej Osinski, Fei Xia, Brian Ichter, and Sergey Levine. Navigation with large language models: Semantic guesswork as a heuristic for planning. In *Proc. of the Conf. on Robot Learning (CoRL)*, pages 2683–2699, 2023.

[104] Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. Cycle-Consistency for Robust Visual Question Answering. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 6642–6651, 2019.

[105] William Shen, Danfei Xu, Yuke Zhu, Li Fei-Fei, Leonidas Guibas, and Silvio Savarese. Situational fusion of visual representation for visual navigation. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, pages 2881–2890, 2019.

[106] Jianbo Shi and Tomasi. Good features to track. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 593–600, 1994.

[107] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[108] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting. *IEEE Trans. on Information Theory*, 58(5):3250–3265, 2012.

[109] Sanjana Srivastava, Chengshu Li, Michael Lingelbach, Roberto Martín-Martín, Fei Xia, Kent Elliott Vainio, Zheng Lian, Cem Gokmen, Shyamal Buch, Karen Liu, et al. Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments. In *Proc. of the Conf. on Robot Learning (CoRL)*, pages 477–490, 2022.

[110] C. Stachniss. *Exploration and Mapping with Mobile Robots*. PhD thesis, University of Freiburg, Department of Computer Science, 2006.

[111] C. Stachniss and W. Burgard. Mapping and Exploration with Mobile Robots using Coverage Maps. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 476–481, Las Vegas, NV, USA, 2003.

[112] Jingwen Sun, Jing Wu, Ze Ji, and Yu-Kun Lai. A survey of object goal navigation. *IEEE Trans. on Automation Science and Engineering*, pages 1–17, 2024.

[113] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training Home Assistants to Rearrange their Habitat. In *Proc. of the Conf. Neural Information Processing Systems (NIPS)*, 2021.

[114] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, **L** ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Proc. of the Conf. Neural Information Processing Systems (NIPS)*, volume 30. Curran Associates, Inc., 2017.

[115] Chaoqun Wang, Jiyu Cheng, Wenzheng Chi, Tingfang Yan, and Max Q.-H. Meng. Semantic-Aware Informative Path Planning for Efficient Object Search Using Mobile Robot. *IEEE Trans. on Systems, Man, and Cybernetics*, 51(8):5230–5243, 2021.

[116] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 7464–7475, 2023.

[117] Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. Hierarchical Open-Vocabulary 3D Scene Graphs for Language-Grounded Robot Navigation. *Proc. of Robotics: Science and Systems (RSS)*, 2024.

[118] Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. DD-PPO: Learning Near-Perfect PointGoal Navigators from 2.5 Billion Frames. In *Proc. of the Intl. Conf. on Learning Representations (ICLR)*, 2020.

[119] Jianzong Wu, Xiangtai Li, Shilin Xu, Haobo Yuan, Henghui Ding, Yibo Yang, Xia Li, Jiangning Zhang, Yunhai Tong, Xudong Jiang, Bernard Ghanem, and Dacheng Tao. Towards Open Vocabulary Learning: A Survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 46(7):5092–5113, 2024.

[120] Xiaojun Xu, Xinyun Chen, Chang Liu, Anna Rohrbach, Trevor Darrell, and Dawn Song. Fooling Vision and Language Models Despite Localization and Attention Mechanism. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4951–4961, 2018.

[121] Karmesh Yadav, Jacob Krantz, Ram Ramrakhya, Santhosh Kumar Ramakrishnan, Jimmy Yang, Austin Wang, John Turner, Aaron Gokaslan, Vincent-Pierre Berges, Roozbeh Mootaghi, Oleksandr Maksymets, Angel X Chang, Manolis Savva, Alexander Clegg, Devendra Singh Chaplot, and Dhruv Batra. Habitat Challenge 2023. `https://aihabitat.org/challenge/2023/`, 2023.

[122] Karmesh Yadav, Arjun Majumdar, Ram Ramrakhya, Naoki Yokoyama, Alexei Baevski, Zsolt Kira, Oleksandr Maksymets, and Dhruv Batra. OVRL-V2: A simple state-of-art baseline for ImageNav and ObjectNav. *arXiv Preprint , arXiv:2303.07798*, 2023.

[123] Karmesh Yadav, Santhosh Kumar Ramakrishnan, John Turner, Aaron Gokaslan, Oleksandr Maksymets, Rishabh Jain, Ram Ramrakhya, Angel X Chang, Alexander Clegg, Manolis Savva, Eric Undersander, Devendra Singh Chaplot, and Dhruv Batra. Habitat Challenge 2022. `https://aihabitat.org/challenge/2022/`, 2022.

[124] Karmesh Yadav, Ram Ramrakhya, Arjun Majumdar, Vincent-Pierre Berges, Sachit Kuhar, Dhruv Batra, Alexei Baevski, and Oleksandr Maksymets. Offline visual representation learning for embodied navigation. In *Proc. of the Intl. Conf. on Learning Representations (ICLR)*, 2023.

[125] B. Yamauchi. Frontier-based exploration using multiple robots. In *International Conference on Autonomous Agents*, pages 47–53, 1998.

[126] Brian Yamauchi. A frontier-based approach for autonomous exploration. In *Proc. of the IEEE Intl. Symp. on Computer Intelligence in Robotics and Automation (CIRA)*, pages 146–151, 1997.

[127] J. Ye, D. Batra, A. Das, and E. Wijmans. Auxiliary Tasks and Exploration Enable ObjectGoal Navigation. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, pages 16097–16106, Los Alamitos, CA, USA, oct 2021. IEEE Computer Society.

[128] Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. VLFM: Vision-Language Frontier Maps for Zero-Shot Semantic Navigation. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2024.

[129] Ryo Yonetani, Tatsunori Taniai, Mohammadamin Barekatain, Mai Nishimura, and Asako Kanezaki. Path Planning using Neural A* Search. In Marina Meila and Tong Zhang, editors, *Proc. of the Intl. Conf. on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*. PMLR, 18–24 Jul 2021.

[130] Bangguo Yu, Hamidreza Kasaei, and Ming Cao. L3MVN: Leveraging Large Language Models for Visual Target Navigation. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 3554–3560, 2023.

[131] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Open-Vocabulary DETR with Conditional Matching. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2022.

[132] Yunqing Zhao, Tianyu Pang, Chao Du, Xiao Yang, Chongxuan Li, Ngai man Cheung, and Min Lin. On Evaluating Adversarial Robustness of Large Vision-Language Models. In *Proc. of the Conf. Neural Information Processing Systems (NIPS)*, 2023.

[133] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, and Jianfeng Gao. RegionCLIP: Region-based Language-Image Pretraining. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 16772–16782, 2022.

[134] Kaiwen Zhou, Kaizhi Zheng, Connor Pryor, Yilin Shen, Hongxia Jin, Lise Getoor, and Xin Eric Wang. ESC: exploration with soft commonsense constraints for zero-shot object navigation. In *Proc. of the Intl. Conf. on Machine Learning (ICML)*. JMLR.org, 2023.

[135] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional Prompt Learning for Vision-Language Models. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[136] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to Prompt for Vision-Language Models. *Intl. Journal of Computer Vision (IJCV)*, 2022.

[137] Zhou, Siyu and Liu, Xin and Xu, Yingfu and Guo, Jifeng. A Deep Q-network (DQN) Based Path Planning Method for Mobile Robots. In *Proc. of the Intl. Conf. on Information and Automation (ICIA)*, pages 366–371, 2018.

List of Figures

List of Tables

List of Algorithms

Appendix A

Poster

Active Perception and Mapping for Open Vocabulary Object Goal Navigation

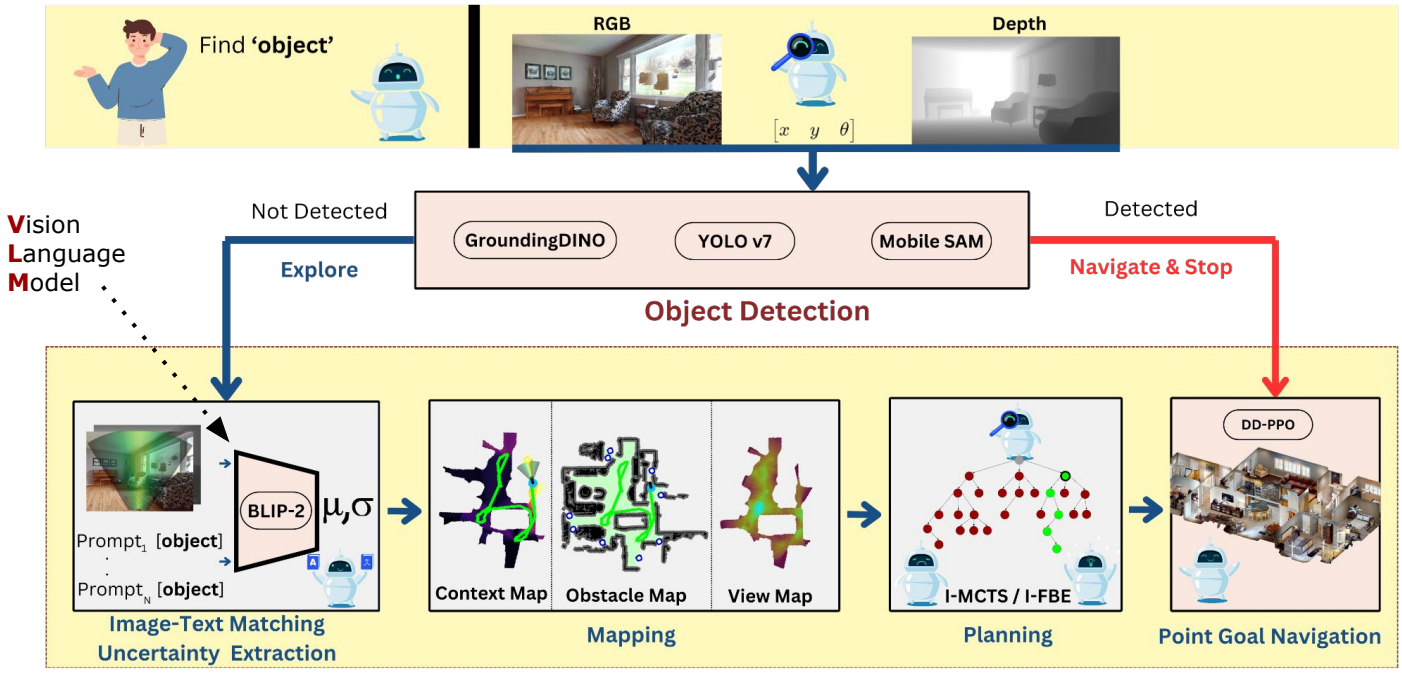
UNIVERSITÄT BONN

Utkarsh Bajpai, Julius Rückin, Marija Popović and Cyrill Stachniss

Abstract

- Household robots need reliable **object-finding** capability
- VLMs enable semantic exploration with some uncertainty.
- We quantify VLM **uncertainty** using contextually similar text prompts & build probabilistic **metric-semantic map**
- Map guides **planning** with frontier-based and tree search methods.

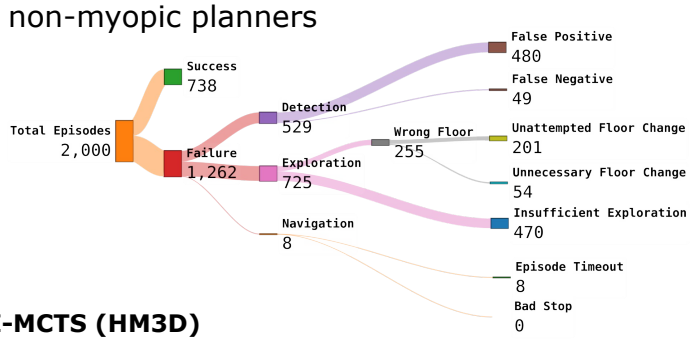
Approach



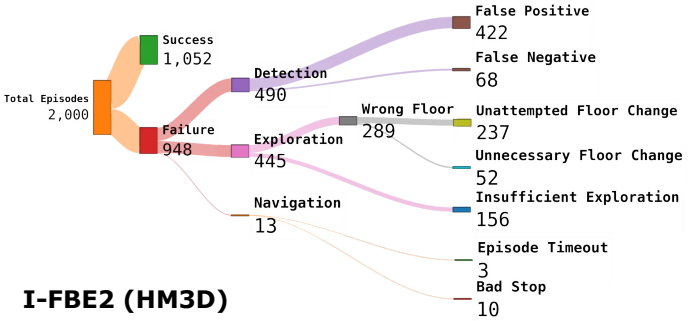
Experiments and Results

- We develop an uncertainty-informed semantic active perception pipeline for object goal navigation
- We develop a probabilistic map formulation to create an uncertainty-infused semantic map representation using VLM uncertainties.
- We develop information-theoretic rewards for myopic and non-myopic planners

Approach	Year	HM3D		MP3D	
		SR	SPL	SR	SPL
ZSON	2023	25.5	12.6	15.3	4.8
CoW	2023	-	-	7.4	3.7
ESC	2023	39.2	22.3	28.7	14.2
VLFM	2024	**52.6**	**30.4**	**36.4**	**17.5**
Ours (I-MCTS)	2024	36.9	16.08	26.9	11.4
Ours (I-FBE1)	2024	52.2	28.9	35.2	16.4
Ours (I-FBE2)	2024	**52.6**	26.8	35.81	16.63



I-MCTS (HM3D)



I-FBE2 (HM3D)