# Combining Learning and Planning for Autonomous Search in Unknown Environments

Lodel, M.

**DOI**
10.4233/uuid:c1146784-04c2-4262-8666-73e8bd1ce4eb

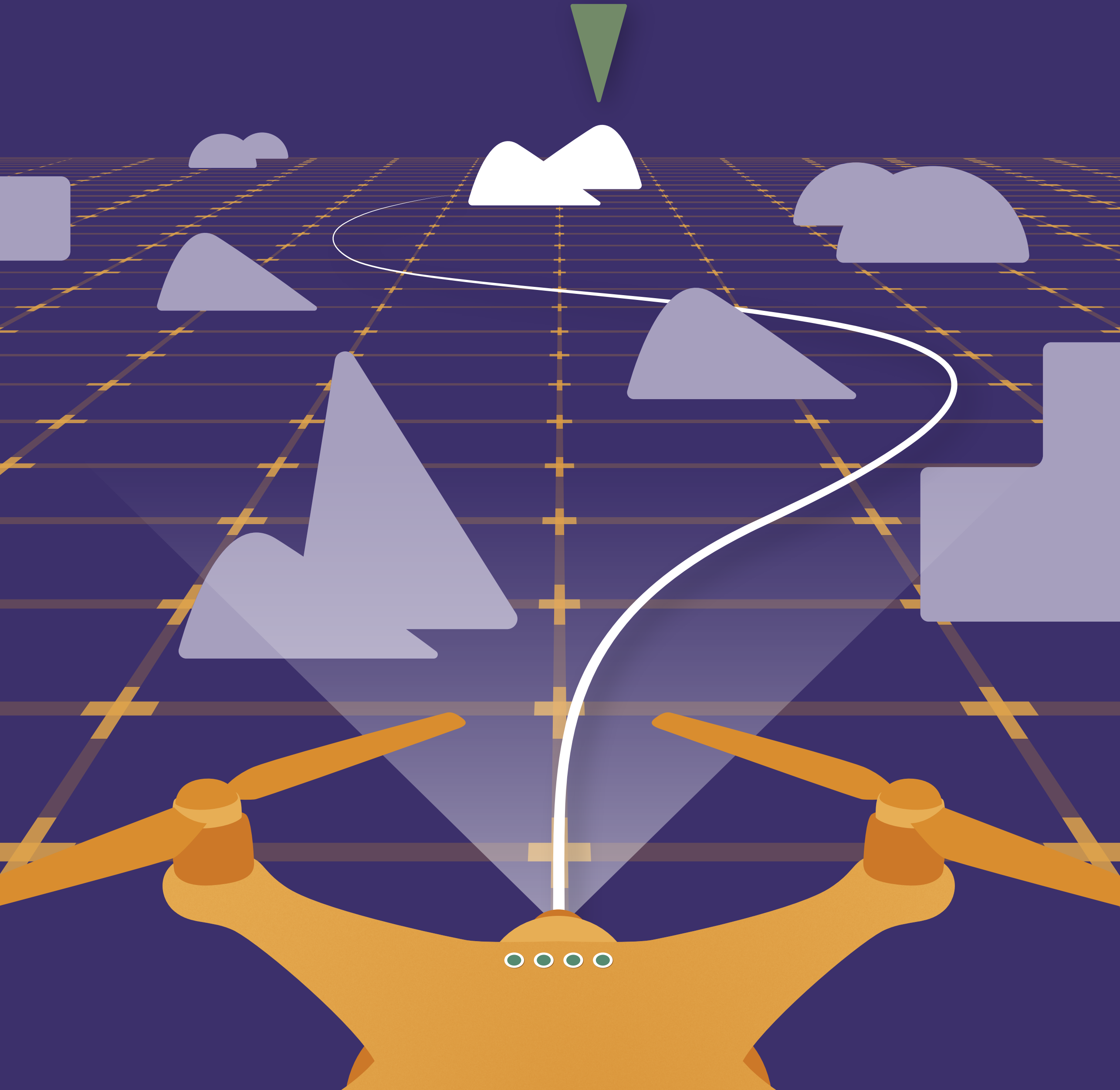
**Publication date**
2025

**Document Version**
Final published version

**Citation (APA)**
Lodel, M. (2025). *Combining Learning and Planning for Autonomous Search in Unknown Environments*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:c1146784-04c2-4262-8666-73e8bd1ce4eb

# Combining Learning and Planning for Autonomous Search in Unknown Environments

Max Lodel



# Combining Learning and Planning for Autonomous Search in Unknown Environments

# Combining Learning and Planning for Autonomous Search in Unknown Environments

## Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus prof. dr. ir. T.H.J.J. van der Hagen,
Chair of the Board for Doctorates,
to be defended publicly on
Monday, 24 November 2025 at 10:00 o'clock

by

**Max LODEL**

Diplom-Ingenieur in Materials and Components for Vehicles,
Technische Universität Bergakademie Freiberg, Germany,
born in Dresden, Germany.

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

| | |
|---|---|
| Rector Magnificus, | voorzitter |
| Prof. dr. J. Alonso-Mora, | Delft University of Technology, *promotor* |
| Prof. dr. R. Babuška, | Delft University of Technology, *promotor* |
| Dr. L. Ferranti, | Delft University of Technology, *copromotor* |
| *Independent members:* | |
| Prof. dr. ir. R. Happee, | Delft University of Technology |
| Prof. dr. ing. F.C. Nex, | University of Twente |
| Prof. dr. K. Alexis, | NTNU Trondheim, Norway |
| Dr. rer. nat. M. Popovic, | Delft University of Technology |
| Prof. dr. ir. M. Wisse, | Delft University of Technology, *reserve member.* |

This research was supported by the National Police of the Netherlands. All content represents the opinion of the author(s), which is not necessarily shared or endorsed by their respective employers and/or sponsors.

*Keywords:* Autonomous Exploration, Motion Planning, Reinforcement Learning, Semantic Scene Understanding, Learning from Human Inputs, Combinatorial Planning, Autonomous Search and Rescue

*Printed by:* Gildeprint | www.gildeprint.nl

*Cover Design:* The Daydreamer's Tale | Samantha Alves Martinho

ISBN 978-94-6518-149-3

An electronic version of this dissertation is available at
`http://repository.tudelft.nl/`.

# Contents

# Summary

Public safety and emergency response agencies increasingly consider the deployment of mobile robots as mounting climate-related disasters and security challenges place human personnel at higher risk and stress. Mobile robots, such as drones, are a promising strategy to respond to these challenges: They can navigate difficult, hazardous terrain, gather real-time situational data, and conduct search or reconnaissance tasks without putting humans at direct risk. However, the currently practiced teleoperation of robots is challenging for such complex missions since the simultaneous navigation, situation assessment, and search tasks can overload human cognitive abilities. Therefore, autonomous planning and decision-making algorithms are required to enable robots to explore and search unknown environments for targets such as missing persons or hazardous materials.

Moving towards this goal, this thesis addresses two core problems. First, local motion planning must carefully account for information gained from sensor observations as well as collision avoidance and the robot's dynamics while moving through cluttered, unknown areas. Second, global exploration planning must strategically select where in the environment to explore to find the target quickly—especially when the environment is large or complex. Given that human operators often possess semantic knowledge about likely target locations, we hypothesize that incorporating such guidance by observed semantic features (e.g., object or room types) into the exploration planning is crucial for time-efficient autonomous search. We address these two core problems by making the following contributions.

The first contribution of the thesis is an informative local motion planning approach that generates safe, collision-free trajectories around obstacles while minimizing uncertainty about the target locations. The critical challenge is to achieve computationally efficient planning of trajectories that maximize information gain under the robot's kinodynamic constraints. In the proposed approach, a model predictive control (MPC) motion planner is guided by a learned viewpoint policy. The policy is trained via deep reinforcement learning (DRL) to maximize long-term information gain by providing a local subgoal to the MPC. The MPC follows the subgoal and ensures that the motion plan remains feasible and collision-free. Therefore, the robot can rapidly replan safe and informative local trajectories online. Simulation experiments demonstrate that the method achieves competitive performance in locating targets compared to a computationally expensive state-of-the-art planner using Monte Carlo Tree Search (MCTS), while allowing for significantly faster execution and replanning.

While local informative planning is crucial for exploring cluttered spaces, it often behaves myopically and inefficiently with respect to large and complex environments. Therefore, the second contribution introduces a global target search planner that balances directed search towards semantically promising areas with complete search space coverage. This planner extends the idea of frontier exploration - focusing observations on the boundaries between explored and unexplored regions - to target search, where different frontiers are assigned a semantic priority. This priority represents the semantic relationships between the target and nearby objects. To minimize target search time, the target search planner schedules high-priority frontiers earlier by solving a custom combinatorial optimization problem to determine the visitation order. By integrating coverage gains into the frontier priorities, the planner ensures that the robot explores the environment efficiently while focusing on semantically relevant areas. We demonstrate this approach in two studies outlined below.

Large, high-quality datasets for learning target-specific semantic relationships are scarce in many real-world scenarios, especially in search and rescue. The third contribution addresses this limitation by proposing a method to learn semantic priority models from expert feedback. Rather than collecting massive amounts of labeled data, the approach exploits an expert operator's sparse guidance inputs in a few target search scenarios. This expert guidance selects a frontier to explore next, which is stored in a training dataset together with the frontier's semantic features. An expert model is then trained to approximate a priority function that predicts how relevant each frontier is for the expert. By incorporating this learned priority function into the global target search planner, the robot can autonomously prioritize semantically relevant areas according to the expert's semantic knowledge. Experiments show that using a small number of expert demonstrations is sufficient for the robot to significantly improve its search efficiency and reduce travel distance until the target is found.

Lastly, the thesis extends semantic target search to three-dimensional environments by integrating it into a 3D planning pipeline for micro aerial vehicles (MAV). The pipeline first detects objects in the environment using onboard vision and associates them with priority values computed from pre-trained large language model (LLM) embeddings. These priorities are then propagated into frontiers in a 3D voxel map, indicating frontier regions that are most likely to contain the target. This enables the evaluation of frontier viewpoints for their information gain that accounts for both semantic priority and volumetric coverage. The viewpoint gains are then used in the combinatorial target search planner to prioritize the viewpoints that most likely lead to the target while ensuring efficient coverage of the environment. By integrating the MAV's kinodynamic constraints into the planning costs, the system ensures smooth, feasible trajectories in real-time. Simulation studies reveal that semantically guided exploration leads to faster and more reliable target discovery than different purely coverage-based exploration baselines. Experiments with a real MAV in the lab confirm the approach's ability to autonomously navigate an MAV through complex 3D environments to a target, exploiting semantic cues, maximizing coverage, and avoiding collisions.

In summary, this thesis demonstrates how planning and learning techniques can be combined for autonomous target search and exploration. These techniques enable mobile robots to navigate unknown environments efficiently and safely while searching for targets and collecting required information. Crucially, our proposed method for semantically guided frontier planning bridges the gap between recent learning-based navigation approaches and established planning-based approaches suitable for real-world robotic systems. By integrating semantic knowledge into robotic exploration, the proposed methods can reduce human operator cognitive load and, therefore, facilitate robot deployment in scenarios such as search and rescue or reconnaissance missions.

# Samenvatting

Openbare veiligheids- en hulpdiensten overwegen steeds vaker de inzet van mobiele robots, omdat toenemende klimaatgerelateerde rampen en veiligheidsuitdagingen menselijk personeel aan grotere risico's en stress blootstellen. Mobiele robots, zoals drones, zijn een veelbelovende strategie om op deze uitdagingen te reageren: ze kunnen navigeren op moeilijk, gevaarlijk terrein, realtime situatiegegevens verzamelen en zoek- of verkenningsopdrachten uitvoeren zonder mensen in direct gevaar te brengen. De huidige teleoperatie van robots is echter een uitdaging voor dergelijke complexe missies, omdat het gelijktijdig uitvoeren van navigatie-, situatiebeoordelings- en zoektaken de cognitieve vermogens van mensen kan overbelasten. Daarom zijn autonome plannings- en besluitvormingsalgoritmen nodig om robots in staat te stellen onbekende omgevingen te verkennen en te doorzoeken op zoek naar doelwitten zoals vermiste personen of gevaarlijke stoffen.

Om dit doel te bereiken, behandelt dit proefschrift twee kernproblemen. Ten eerste moet lokale bewegingsplanning zorgvuldig rekening houden met informatie die is verkregen uit sensorwaarnemingen, evenals met het vermijden van botsingen en de dynamica van de robot tijdens het bewegen door rommelige, onbekende gebieden. Ten tweede moet bij de planning van de globale verkenning strategisch worden gekozen welke delen van de omgeving moeten worden verkend om het doel snel te vinden, vooral wanneer de omgeving groot of complex is. Aangezien menselijke operators vaak semantische kennis hebben over waarschijnlijke doellocaties, veronderstellen we dat het opnemen van dergelijke begeleiding door waargenomen semantische kenmerken (bijv. object- of kamertypes) in de verkenningsplanning cruciaal is voor een tijdsefficiënte autonome zoektocht. We pakken deze twee kernproblemen aan door de volgende bijdragen te leveren.

De eerste bijdrage van het proefschrift is een informatieve lokale bewegingsplanningsaanpak die veilige, botsingsvrije trajecten rond obstakels genereert en tegelijkertijd de onzekerheid over de doellocaties minimaliseert. De cruciale uitdaging is om een rekenkundig efficiënte planning van trajecten te realiseren die de informatieverzameling maximaliseren binnen de kinodynamische beperkingen van de robot. In de voorgestelde aanpak wordt een bewegingsplanner op basis van model predictive control (MPC) geleid door een aangeleerde policy. De policy wordt getraind via deep reinforcement learning (DRL) om de informatieopbrengst op lange termijn te maximaliseren door een lokaal subdoel aan de MPC te geven. De MPC volgt het subdoel en zorgt ervoor dat het bewegingsplan haalbaar en botsingsvrij blijft. Daardoor kan de robot snel veilige en informatieve lokale trajecten online herplannen. Simulatie-experimenten tonen aan dat de methode concurrerende prestaties levert bij het lokaliseren van doelen in vergelijking met een rekenkundig dure state-of-the-art planner die gebruikmaakt van Monte Carlo Tree Search (MCTS), terwijl de uitvoering en herplanning aanzienlijk sneller verlopen.

Hoewel lokale informatieve planning cruciaal is voor het verkennen van rommelige ruimtes, werkt deze vaak kortzichtig en inefficiënt in grote en complexe omgevingen. Daarom introduceert de tweede bijdrage een globale doelzoekplanner die een evenwicht

vindt tussen gericht zoeken naar semantisch veelbelovende gebieden en volledige dekking van de zoekruimte. Deze planner breidt het idee van grensverkenning – waarbij observaties worden gericht op de grenzen tussen verkende en onverkende gebieden – uit naar doelzoekopdrachten, waarbij aan verschillende grenzen een semantische prioriteit wordt toegekend. Deze prioriteit vertegenwoordigt de semantische relaties tussen het doel en nabijgelegen objecten. Om de zoektijd naar het doel te minimaliseren, plant de planner grenzen met hoge prioriteit eerder in door een aangepast combinatorisch optimalisatieprobleem op te lossen om de volgorde van bezoeken te bepalen. Door dekkingswinst te integreren in de grensprioriteiten, zorgt de planner ervoor dat de robot de omgeving efficiënt verkent en zich tegelijkertijd concentreert op semantisch relevante gebieden. We demonstreren deze aanpak in twee studies die hieronder worden beschreven.

Grote, hoogwaardige datasets voor het leren van doelgerichte semantische relaties zijn schaars in veel praktijksituaties, vooral bij zoek- en reddingsoperaties. De derde bijdrage pakt deze beperking aan door een methode voor te stellen om semantische prioriteitsmodellen te leren op basis van feedback van experts. In plaats van enorme hoeveelheden gelabelde gegevens te verzamelen, maakt deze aanpak gebruik van de schaarse begeleiding van een deskundige operator in een paar doelzoekscenario's. Deze deskundige selecteert een grensgebied om vervolgens te verkennen, dat samen met de semantische kenmerken van het grensgebied wordt opgeslagen in een trainingsdataset. Vervolgens wordt een model getraind om een prioriteitsfunctie te benaderen die voorspelt hoe relevant elke grens voor de deskundige is. Door deze aangeleerde prioriteitsfunctie op te nemen in de globale doelzoekplanner, kan de robot autonoom semantisch relevante gebieden prioriteren op basis van de semantische kennis van de deskundige. Experimenten tonen aan dat het gebruik van een klein aantal demonstraties door deskundigen voldoende is om de zoekefficiëntie van de robot aanzienlijk te verbeteren en de afgelegde afstand tot het doel te verkleinen.

Ten slotte breidt het proefschrift het semantisch zoeken naar doelen uit naar 3D omgevingen door het te integreren in een planningpijplijn voor micro aerial vehicles (MAV), oftewel kleine luchtvaartuigen. De pijplijn detecteert eerst objecten in de omgeving met behulp van beeldverwerking en koppelt deze aan prioriteitswaarden die zijn berekend op basis van embeddings van een vooraf getraind large language model (LLM). Deze prioriteiten worden doorgegeven aan grenzen in een 3D-voxelkaart, waarmee de grensgebieden worden aangegeven die het meest waarschijnlijk het doel bevatten. Dit maakt het mogelijk om grensgebieden te evalueren op hun informatiewinst, waarbij zowel rekening wordt gehouden met semantische prioriteit als volumetrische dekking. De waarde van mogelijke uitzichtpunten wordt gebruikt in de combinatorische doelzoekplanner om prioriteit te geven aan de grensgebieden die het meest waarschijnlijk naar het doel leiden, terwijl een efficiënte dekking van de omgeving wordt gegarandeerd. Door de kinodynamische beperkingen van de MAV te integreren in de planningskosten, zorgt het systeem voor soepele, haalbare trajecten in realtime. Simulatiestudies tonen aan dat semantisch gestuurde verkenning leidt tot snellere en betrouwbaardere doelontdekking dan verschillende alleen op dekking gebaseerde verkenningsbaselines. Experimenten met een echte MAV in het lab bevestigen verder het vermogen van de aanpak om een MAV autonoom door complexe 3D-omgevingen naar een doel te navigeren, waarbij semantische aanwijzingen worden gebruikt, de dekking wordt gemaximaliseerd en botsingen worden vermeden.

Samenvattend laat dit proefschrift zien hoe planning- en leertechnieken kunnen wor-

den gecombineerd voor het autonoom zoeken naar en verkennen van doelen. Deze technieken stellen mobiele robots in staat om efficiënt en veilig door onbekende omgevingen te navigeren terwijl ze naar doelen zoeken en de benodigde informatie verzamelen. Cruciaal is dat onze voorgestelde methode voor semantisch gestuurde grensplanning de kloof overbrugt tussen recente, op leren gebaseerde navigatiebenaderingen en gevestigde, op planning gebaseerde benaderingen die geschikt zijn voor roboticasystemen in de praktijk. Door semantische kennis te integreren in robotica-exploratie, kunnen de voorgestelde methoden de cognitieve belasting van menselijke operators verminderen en daarmee de inzet van robots in scenario's zoals zoek- en reddingsacties of verkenningsmissies vergemakkelijken.

# Acknowledgments

More than four years of a PhD project and living in a new country is nothing less than a rollercoaster ride, but luckily I was not alone on this ride. Here, I would like to express my gratitude to all the people that supported me and that shaped my experience and my memories of this unforgettable, amazing time.

First, I want to thank my main supervisor and promotor, Javier, for guiding me through the journey all the way. Your patience and optimistic mindset were important to helping me learn and grow, and to achieve the results of this thesis and our project. I was lucky to have two more supervisors in our project. Robert, thank you especially for the open conversations not only about the content of my research but also about the process in general, which helped me put things in perspective. Laura, I really appreciated your effort to emphasize positive and motivating thoughts, which I believe is very valuable in our fast-paced and sometimes overwhelming research field.

It's also been a pleasure to collaborate with the team at the Dutch Police, Klaas Jan, Paul, and Marius. Thank you for the insightful meetings, where I learned much about the practical challenges with deploying robots, and especially thank you for the sincere encouragement to learn and speak Dutch with you. Dank jullie wel!

A special thanks goes to Dennis, my partner-in-crime in the project with the police. It's always been fun to chat during our many train rides to project meetings, and I think we can be both proud of what we put together for the project demos and the lab setup! To Thijs, thank you for the great work on the drone setup and for the good mood when preparing the demo. I also want to express my gratitude to Bruno for helping me to get started with my research in my first year, and to Nils, who provided a lot of guidance on how to formalize and structure my research ideas in the second half of my PhD.

While the PhD research was challenging and exhausting at times, all the nice people at CoR have been creating a positive environment that made this time nevertheless very enjoyable, both at work and outside. Lasse, thank you for your openness and the inspiring conversations we had over craft beer, good coffee or other culinary experiences of which Delft has plenty to offer. Álvaro, thank you for always being encouraging in our long discussions about our research and PhD life. Many thanks also to Elia, Corrado, Rodrigo, Giovanni, Italo, and Tomás for being a great community beyond work and all the good times together at after-lunch coffees, after-work beers, dinners, parties, and the climbing gym. To Luzia, Andreu, Saray, Max S., Max K., Linda, Anna, Tom, and the other colleagues at AMR and CoR - Thank you for insightful discussions and the good atmosphere at lunch breaks together. Thanks to Nikhil for challenging some of my ideas and building on top of others in your Master's thesis, which contributed a lot to this PhD project.

Moving to a different country in the middle of the pandemic could have been very tough (even without starting a PhD), but I was incredibly lucky to share a house with the best people, who made the lockdown time unforgettable in a good way, from everyday dinners together to theme parties - big big thanks to the Geerweg family: Anna, Naga, Uttam, Māra, Ivneet, Aleksandra, Mateusz, Zeynep, and Lotte. To Anna, for being one of my best friends in Delft, you would always take time to talk about whatever I was dealing with, and give me new perspectives. Also, thank you for the great time on hikes around the Netherlands. To Naga, Uttam, and Ivneet, for your contagious positivity at all times, for making sure I was always well fed with the most amazing Indian food, and for sharing your culture so openly with us at Diwali dinners and during our amazing stay in Bengaluru. Big thanks also to the Foulkeslaan people, Paul, Maite, Guti, Blanca, Sam, Lucas, and Jonas, for the wonderful time together during dinners and barbecues, and for teaching me that a presentation night can be a hilarious activity. To Paul and Maite, thank you for the fun and cozy weekends together, even if they involved rollercoasters. To Sam and Niels, thank you for being the best neighbors, for hours-long conversations over comfy dinners, to Niels for the regular fresh bread supplies, and to Sam for designing the amazing cover of this thesis. Thanks to Bea and María, for the good time together in the Netherlands, Belgium and Portugal, and for the inspiring discussions about (PhD) life.

Starting a new life abroad and meeting so many new people, however, also means having less time with the people who have been important to me for a long time. Nevertheless, all of you in Germany stayed part of my life, and every time we managed to meet, in Dresden, on group vacations or when some of you visited me in Delft, it made me incredibly happy to know that no matter what, you would always be there. To Rico, Markus, Clara, Clemens, Caro, Henni, Georg, Flo, Fine, Hannah, Ludwig, Benny, Benni, Kathi, Marius, Jasper, Marie, Liesa, Lucas, and Hanna, thank you all so much for your support! A special thanks to Rico, who visited me in Delft uncountable times, which especially during the first year meant a lot to me.

Agradeço também à minha família portuguesa, em especial à Sheila e ao Nuno, por me terem apoiado e motivado ao longo destes anos. Graças a vocês, Lisboa tornou-se a minha segunda casa, onde me sinto (e como) sempre tão bem. Muito obrigado!

Der Weg, der mir diese tolle Erfahrung, in den Niederlanden zu leben und dort zu promovieren, ermöglicht hat, begann natürlich schon viel, viel früher. Daher möchte ich ganz herzlich den Menschen danken, die diesen Weg schon seit über drei Jahrzehnten geebnet haben: meine Familie. Ganz besonders danke ich meinen Eltern: Ihr habt nie an mir gezweifelt, meine Interessen, so oft sie sich auch gewandelt haben, immer unterstützt und durch euer Engagement für mich konnte ich all das lernen und erfahren, was mein Leben heute so spannend und schön macht. An meinen Bruder Felix, danke, dass du mir schon lange vor Augen geführt hast, dass es nicht nur in Wissenschaft und Technik, sondern auch in der Kunst viel zu entdecken und zu bestaunen gibt, auch wenn meine Einsicht etwas gebraucht hat. An meine Oma in Dresden sowie Oma und Opa in Königstein, danke, dass ihr immer für mich da wart und mich unterstützt habt, und mir die Neugier und den Ehrgeiz mitgegeben habt, die mich bis hierher getragen haben.

The most important person on this crazy rollercoaster ride has been, of course, my girlfriend Beatriz, who sat by my side for every new loop. Bea, thank you so much for being up for these adventures – from starting a long-distance relationship in Erasmus to moving first to the Netherlands, and now to Germany. More than anything else, your support and love every day have given me the energy to tackle every challenge – during the PhD and today. I cannot wait to see what our adventures will bring next! Amo-te!

*Max, October 2025*

# 1

# Introduction

*This chapter motivates the presented research from the societal context, emphasizing how autonomous robots can support public safety agencies to face mounting challenges and protect their personnel. Focusing on search and rescue use cases, we identify two core challenges for safe and efficient autonomous robot planning in unknown environments: Informative local motion planning and semantically-driven global exploration planning. The chapter further outlines our approach to these challenges in relation to existing methodologies and summarizes the contributions of this thesis.*

Public safety and emergency response agencies face mounting challenges due to the increasing frequency and severity of climate change-related natural disasters, such as floods, storms, and wildfires. At the same time, they must contend with complex human-made threats to society, including terrorism and organized crime. These developments place law enforcement officers and emergency responders at high risk and strain, requiring them to operate intensively in hazardous, dangerous environments and threatening their physical and mental health. Human physical limitations can constrain the effectiveness of the response to these scenarios. Naturally, there is a reluctance to deploy humans into hazardous conditions, which potentially delays a response to time-critical situations. Moreover, human capabilities are limited, for example, in navigating tight or unstable terrains and in rapidly searching large, cluttered spaces. Finally, demographic change will reduce the availability of human responders, exacerbating many of the mentioned challenges.

With recent advances in robot hardware and control [1, 2], computer vision [3–5], and embodied artificial intelligence [6–8], autonomous robots promise a new avenue for public safety agencies to respond to these challenges. Deploying autonomous robots in these scenarios can alleviate human limitations and enable more successful missions, as they can access difficult terrains and gather information from large, complex scenes while, most importantly, not putting humans at risk. For example, micro aerial vehicle (MAVs) equipped with onboard cameras can be small enough to search collapsed buildings for survivors but can also rapidly cover large outdoor areas to locate missing persons. Moreover, MAVs, wheeled robots, or legged robots can take over repetitive, resource-intensive tasks, such as securing sensitive locations, which frees up human responders for other, non-automatable tasks.

In this thesis, we focus on the use of autonomous robots for search and rescue and reconnaissance missions, where the robots are tasked with exploring an unknown environment. Specifically, we consider that the mission goal is to locate one or more targets of interest in the scene, such as missing persons or hazardous materials. Such *target search* missions are characterized by the need for the robot to safely and efficiently navigate through environments with many obstacles and complex structures while gathering information about the scene to locate the target. However, current robot deployments in such missions rely on skilled human operators to teleoperate the robots. Facing the challenges of unknown environments and uncertainty about the scenario, the human operator must guide the robot safely through the scene while simultaneously interpreting the robot's camera images to assess the situation and locate the target. The complexity and variety of these cognitive tasks have the potential to overwhelm the human operator, constraining the efficiency and effectiveness of robot deployment in complex missions. Therefore, there is a need to equip robots with the autonomous capabilities required in search missions to reduce the cognitive load on human operators.

## 1.1 Problem Overview

To operate autonomously in a target search mission, a robot must be able to *perceive* its environment, *plan* its next actions, and *control* its motion. The focus of this thesis is on the robot's planning methodologies, that reason about *where* and *how* to move in the environment to find the target quickly. Planning in autonomous target search missions in unknown environments is challenging due to the uncertainty about the scene's geometric structure and the location of the target. Leveraging partial information, the robot must plan its actions to reduce this uncertainty by gathering information about the scene. This thesis addresses two main planning problems within these challenges: Local motion planning that considers the information gained along the trajectory, and global exploration planning that efficiently searches the environment for the target.

Local motion planning generates collision-free trajectories that adhere to the robot's kinematic constraints. As the robot makes observations when moving along the trajectory, the trajectories should lead to informative observations that efficiently reduce the uncertainty about the scene. Moreover, when moving around obstacles in densely cluttered areas, planning to gather information from occluded areas is crucial. Considering collision avoidance, the robot's dynamics and possible future observations are especially challenging with limited onboard computational resources.

When searching large and complex environments, such local planning methods alone are likely to lead to inefficient behavior, and the robot requires a global planning strategy that decides where to explore next to find the target quickly. Importantly, exploration should be directed toward the most promising areas for the search of the target. While pure exploration strategies [9–11] efficiently cover the entire environment, a human operator would steer the robot towards areas where the target is likely to be found. Human operators can leverage semantic information, such as detected objects or room types, as well as their experience, to reason about the target's likely location. For example, dangerous chemicals are more likely to be found in a storage room than in an office. When handing over control to the robot, human operators expect the robot to incorporate such human semantic priors into its planning strategy. However, such reasoning is challenging due to the uncertainty about the scene and the semantic relationships that can indicate the target's location. Hence, it is crucial that while exploration should be directed, it must also be efficient by avoiding redundant observations and covering all parts of the environment.

In summary, the goal of this dissertation is to *develop planning methodologies that enable autonomous robots to efficiently and safely explore unknown environments to search for targets of interest, while leveraging expert knowledge.*

## 1.2 Approach

This section provides an overview of the approaches and concepts developed in this thesis to address the mentioned challenges in planning for autonomous target search missions. Planning a sequence of future actions is a key capability for autonomous navigation, as it allows robots to perform consistent behaviors in complex scenarios instead of relying on inefficient reactive or greedy strategies. However, planning in unknown environments is challenging due to the partial observability of the problem, i.e., critical information for reasoning about future actions, such as the environment's structure and semantic features, are not fully known. Explicit planning under partial observability needs to search through and evaluate many possible future states, leading to a combinatorial explosion in the search space, which is computationally infeasible for real-time operation. This motivates a central theme of the approaches presented in this thesis: *Guiding planning methods with learning-based reasoning about higher-level tasks, such as information gathering or semantic scene understanding, can improve the robot's reasoning capabilities in unknown environments.*

### 1.2.1 Informative Local Motion Planning

Informative local motion planning generates feasible trajectories that guide the robot around nearby obstacles, avoiding collisions while maximizing the information gained from the environment. Maximizing future information gain means optimizing poses in the planned trajectories such that observations from those poses are expected to achieve the highest possible reduction in uncertainty about the environment. Evaluating information gain from future observations must take into account the robot's sensor model, observed obstacles to avoid occlusions, and earlier observations to avoid redundant information. The approach presented in this thesis aims to alleviate the high computational burden of evaluating future information gains by learning a policy that guides the local motion planner toward informative poses. Local motion planning is realized using Model Predictive Control (MPC), and the policy is trained using Reinforcement Learning (RL).

**Model Predictive Control** (MPC) generates control inputs by solving a constrained optimization problem at each step, ensuring feasibility under the robot's dynamics and collision avoidance requirements. The objective is typically to track a reference path or goal position, with only the first input applied before re-optimizing based on updated robot and environment states. The advantage of MPC is its ability to generate smooth trajectories respecting safety constraints crucial for real-world deployment and to replan at each time step to account for changing environmental conditions [12–14]. Its limitation for autonomous search and exploration is that complex optimization objectives, such as maximizing the information gain, are too computationally expensive for the large number of evaluations necessary in sampling-based or gradient-based optimization methods.

**Reinforcement Learning** (RL) enables an agent to learn a policy that maximizes a reward signal by training on a dataset of prior experiences. The policy, often represented by neural networks, maps observations to actions, allowing for fast online execution after training. The key advantage of RL is that the computational burden of potentially high-dimensional optimization, such as in MPC, is shifted to the training phase. Additionally, RL maximizes rewards over an infinite time horizon, allowing for learning policies that pursue long-term objectives. RL is often used for decision-making problems under partial observability [6, 13, 15] since the policy is trained to infer the best action just from observations, implicitly reasoning about the environment's hidden state, such as the target location. Deep learning models, such as convolutional neural networks (CNNs) or graph neural networks (GNNs), have enabled RL to learn complex policies from high-dimensional observations, such as images or time series data [6, 16].

### 1.2.2 Global Exploration Planning for Target Search

In time-critical scenarios such as search and rescue missions, long-horizon planning is required to efficiently explore and search complex environments. Conversely, planning methods such as MPC are computationally limited to short-term horizons, resulting in locally optimal trajectories that may lead to inefficient exploration behavior, such as repeatedly visiting the same area or missing important regions of the environment. RL-based methods, on the other hand, can deal with long horizons, but their generalization abilities are constrained by the availability of training data, which is naturally scarce for highly dynamic and uncertain public safety scenarios.

Efficient target search also requires reasoning about observed semantic information, such as detected objects, to guide exploration towards promising areas. Such directed search can be achieved using learning-based methods, capturing the complex semantic relationships [6–8, 16]. However, these methods are also limited by the availability of training data for the targeted scenarios. Faced with unseen scenarios, learned semantic reasoning models may generate suboptimal behavior, such as incomplete exploration, thus lacking robustness for sensitive missions. To maintain robust but directed exploration, this thesis explores guiding long-horizon combinatorial exploration planning with learning-based prediction of target locations from semantic information.

**Combinatorial Exploration Planning** A classical approach for exploring large unknown environments is frontier-based exploration [17], where the robot moves to the nearest frontier, i.e., the boundary between explored and unexplored areas. As this greedy choice can lead to inefficient behavior, planning over a set of frontier locations is a popular approach for exploration [9–11, 18]. Here, the continuous frontiers are discretized, usually by clustering or sampling, and a planner determines the optimal visitation order of the frontiers. The first frontier in the plan is chosen as the next goal for downstream planning, and a new plan is generated once new frontiers are found. Finding an efficient visitation order for a set of locations is a combinatorial problem known as the Traveling Salesman Problem (TSP) and its variants. While these problems are NP-hard, efficient heuristics exist to find near-optimal solutions and allow for real-time planning. This way, long-horizon reasoning can be achieved by planning over a discrete set of relevant observation locations. Combinatorial frontier planning has enabled robust real-world deployment of autonomous robots, e.g., MAVs, for exploration tasks [9, 11]. While this approach can efficiently cover the environment, it lacks reasoning about semantic information that could guide exploration toward promising areas for target search.

**Semantically Guided Target Search** Learning the semantic relationships between observed objects and the target is an effective method to guide a robot toward a target in an unknown environment. Using data from interaction with the environment, guidance policies [6, 16] or cost-to-go functions [19, 20] can be trained to direct the robot toward the target. Alternatively, foundation models such as large language models [21, 22] can leverage their general internet-scale training data to infer promising search locations [7, 8, 23]. However, for highly specialized scenarios such as search and rescue, we cannot expect that sufficient training data is available to learn the semantic relationships. As human operators hold the necessary knowledge about scenario-specific semantics, this thesis explores learning a model of semantic priorities from human inputs. Using techniques from reward learning [24, 25], a priority model can be learned from a few human demonstrations, such that it can guide the robot towards promising areas for target search.

## 1.3 Contributions and Outline

In the following, the main scientific contributions of this thesis are summarized:

(1) **An informative local motion planning** approach to generate safe robot trajectories maximizing information gain about an unknown environment. An MPC-based local motion planner guarantees the safety and feasibility of the generated trajectories, constraining the motion plans to obstacle-free space around the robot. A viewpoint policy is trained using deep RL to guide the MPC planner with a refer-

ence subgoal, such that mutual information from observations along the trajectory is maximized. The policy is modeled using multiple CNN encoders to process 2D map representations of surrounding obstacles and environment uncertainty. The approach is evaluated in simulation, showing competitive performance compared to an information-maximizing planner based on Monte Carlo Tree Search (MCTS), while requiring significantly less online execution time.

(2) **A semantically-guided frontier exploration planner for target search** that prioritizes promising frontiers for finding the target while ensuring efficient and complete exploration of complex environments. Given a semantic priority model and a discrete set of exploration frontiers, the planner generates a visitation order that schedules semantically promising frontiers early in the plan. Specifically, the planner solves a priority-weighted minimum latency problem, penalizing arrival times at frontiers weighted by their priority. By combining semantic priority and coverage information gain in the priority weights, the planner can effectively guide the robot toward the target while ensuring efficient exploration of the environment. This approach has been applied with two different methods of computing priority weights, each tested in a different experimental setting:

   (a) **Semantic priorities learned from expert inputs** that can guide the robot towards areas where the expert expects the target. Assuming that the expert leverages its knowledge and experience about the scenario when guiding the robot by choosing the next frontier to explore, this thesis proposes to learn a semantic model from recorded inputs. An expert model of frontier choices is devised, where a semantic priority function encapsulates the expert's semantic knowledge about the environment, scoring frontiers based on their semantic relevance to the target. Training this function such that the model fits recorded data allows leveraging of the semantic priority function to semantically guide the robot without expert guidance. Simulation experiments in 2D environments with the combinatorial exploration planner show that consistent target search outperforming coverage exploration can be achieved by learning only from a few expert inputs.

   (b) **A 3D target search pipeline for MAVs** that can evaluate the semantic priority of 3D frontier viewpoints for guiding the target search frontier planner. In this work a semantic priority model based on LLM word embeddings is used, evaluating the semantic relationship between any observed object and the target. To scale the target search planner to 3D environments and exploration using a MAV with an onboard camera constrained by its limited field of view, a pipeline to propagate object semantic priorities to frontier viewpoint priorities is developed. To this end, object priorities are diffused into a 3D priority voxel map, and sampled frontier viewpoints are evaluated based on the priority-weighted sum of visible frontier voxels, ensuring that both target search and coverage exploration are pursued. The proposed pipeline is validated in both simulation and hardware experiments, showing that the exploration planner can efficiently guide the MAV through complex environments toward semantically relevant 3D viewpoints that lead to the target.

Chapter 2 presents the developed method for informative local motion planning, as well as the corresponding simulation results. Chapter 3 introduces the method for learning semantic priorities from expert inputs and the first variant of the combinatorial target search planner and provides the simulation results for the combination of both methods. Chapter 4 presents the 3D target search pipeline for MAVs, including the viewpoint evaluation method and a variation of the target search planner, and the simulation and hardware experiments with a MAV. Finally, Chapter 5 concludes the thesis and its key findings and provides an outlook on future research directions.

# 2

# Learning Viewpoint Recommendations for Informative Trajectory Planning

*Informative local motion planning is crucial for ensuring safe autonomous navigation through cluttered spaces while maximizing the information about the environment gained from the robot's onboard sensors. Crucially, a safe and efficient approach needs to replan quickly based on new observations and guarantee collision avoidance. This chapter presents a hierarchical planning framework, where a learned policy network provides short-term viewpoint recommendations that guide an MPC-based trajectory planner. The policy is trained with deep reinforcement learning to provide viewpoint recommendations that maximize cumulative information gain, ensuring non-myopic, efficient information gathering. The MPC planner aims to follow the provided viewpoint and enforces collision-free, dynamically feasible motions. In simulation tests in previously unseen 2D environments, our method consistently outperforms greedy next-best-view policies and achieves competitive performance compared to Monte Carlo Tree Search in terms of cumulative information gain and coverage time, with a reduction in execution time by three orders of magnitude.*

This chapter is based on: ***M. Lodel***, *B. Brito, A. Serra-Gomez, L. Ferranti, R. Babuška, J. Alonso-Mora, "Where to Look Next: Learning Viewpoint Recommendations for Informative Trajectory Planning", 2022 IEEE International Conference on Robotics and Automation (ICRA).*

## 2.1 Introduction

### 2.1.1 Motivation

Autonomous robots can play a fundamental role in gathering information in critical and dynamic scenarios, such as search and rescue [26, 27] or environmental monitoring [28, 29]. For example, robots can support human emergency responders to locate victims in challenging or dangerous terrain. In such scenarios, environments are often unknown, and autonomous navigation methods must continuously replan actions that maximize the information gathered over long horizons. Moreover, these trajectories must be efficient with respect to time or energy costs.

Long horizon, or *non-myopic*, path planning methods for information gathering and map exploration [28, 30–37] suffer from high computational cost and thus long planning times, particularly in complex, obstacle-rich environments. To enable fast online execution, recent works have approached information gathering using deep reinforcement learning (DRL) [15, 27, 38–42]. In these methods, a policy learns in offline training to select an action that maximizes the expected information gain of future observations. The policy is usually modeled as a deep neural network that reasons about high-dimensional observations of the agent's surroundings (e.g., obstacles), or its current belief about the environment. However, DRL-based information gathering methods do not explicitly consider constraints for collision avoidance and do not account for the robot's dynamics.

In uncertain or dynamic scenarios, it is advantageous to employ an optimization-based local motion planner such as model predictive control (MPC), to generate dynamically feasible and collision-free trajectories and thus safe robot motion [12, 43]. The recent work of [13] combined MPC with a learned subgoal policy for navigation among interacting agents. In this paper, we propose a hierarchical framework, depicted in Figure 2.1, for exploring unknown, obstacle-rich environments. Building on the idea of [13], we enhance a local motion planner with a guidance policy trained using DRL. By training in different simulated environments, the DRL agent learns a guidance policy that maximizes information gains from future sensor observations. In particular, the policy is trained to combine its belief about the environment with local observations of obstacles and the robot state for guiding an MPC-based motion planner by recommending a subgoal reference. This *viewpoint reference*, i.e., a subgoal leading towards informative observations, is then used by the MPC planner to generate low-level control commands while ensuring collision avoidance and kinodynamical feasibility of the trajectory.

### 2.1.2 Related Work

#### Planning for Information Gathering

Informative path planning (IPP) methods plan future observation poses that are expected to reduce uncertainty about the environment as efficiently as possible, generally at the expense of computation time. Generally, *myopic* and *non-myopic* IPP methods can be distinguished. Myopic methods capitalize on the submodularity property of common IPP objectives such as maximizing mutual information [44]. These methods select their actions greedily either by considering the next best viewpoint at the current time step [45–47], or by finding a trajectory leading to the best reachable next viewpoint [48]. While their computation times are generally low, these methods sacrifice efficiency in terms of time or

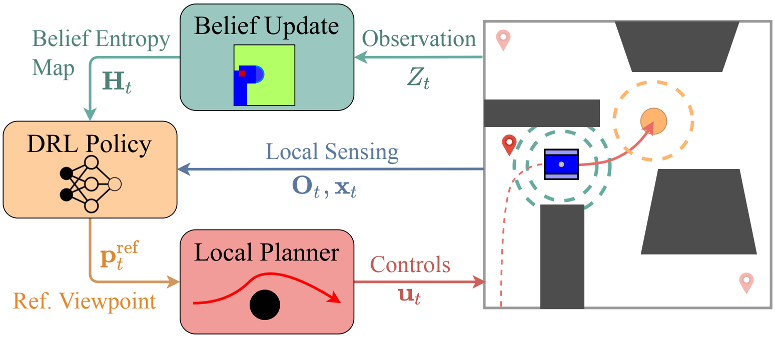


Figure 2.1: Conceptual overview of the proposed informative trajectory planning framework. A DRL policy recommends a viewpoint reference to a local planner, based on the robot's current belief about the environment, and local sensory information. The local planner generates a feasible trajectory and executes control commands, leading the robot (depicted in blue on the right-hand side) to reduce the uncertainty about the environment.

energy required to gather information about the environment due to their short planning horizon.

Non-myopic planning methods, in contrast, attempt to find long-horizon plans that maximize an information-related objective quantifying the cumulative information gain. These methods often rely on tree search algorithms [30–34], such as Monte Carlo Tree Search (MCTS) [30–32], or global optimization [28, 37]. While being able to find near-optimal paths over long horizons, they suffer from high computational costs due to repeated predictions of possible future observations. This is particularly exacerbated by computationally expensive visibility checks in obstacle-rich environments. The resulting long re-planning times make them unpractical for time-constrained and fast-paced dynamic scenarios. These computational issues are partially addressed by [34, 35], but as the aforementioned methods, they simplify or do not consider the kinodynamic constraints of the robot.

Local motion planners can ensure dynamic feasibility and collision avoidance, but this might result in a trajectory deviating from the planned informative path. Maximizing an information-theoretic objective directly in local trajectory optimization has been proposed for SLAM [49, 50] and grid mapping [36]. This approach requires a differentiable information gain model and is computationally expensive for long planning horizons.

Our proposed framework, in contrast, combines fast online execution times with non-myopic reasoning and explicit dynamic feasibility and is flexible with respect to observation and information model design choices. This is enabled by combining local motion planning with DRL.

### DRL for Information Gathering

Thanks to their fast execution times and ability to choose actions conditioned on the recent history of observations, DRL-based approaches have the potential to find a suitable trade-off between quickly reacting to new observations and efficient information gathering. A common component of previous DRL-based information gathering approaches [15, 27, 38–42, 51] is that the agent's current incomplete knowledge about the environment is formulated as an observation to reason about where informative sensor measurements can be taken. The methods differ in the type of actions being selected, and thus the way the policy interacts with the robot. A common approach is to select from a discrete set of motion primitives [15, 38, 41] for a simplified first-order dynamic model, and directly apply them to the robot. In other works, the learned policy makes higher-level decisions (e.g., by choosing the next frontier [27], subgoal [42], subregion [39] or graph vertex [40, 51] to observe). In [27, 39, 42], that action is executed by a lower-level path planner.

However, none of the mentioned works does explicitly account for how the actions chosen by the DRL agent can result in dynamically feasible, collision-free trajectories. Our method trains a policy to give a high-level local subgoal, or *viewpoint* reference, to a lower-level MPC trajectory planner that can ensure the satisfaction of the robot's constraints. In contrast to the global subgoal policy in [42], the subgoals in our method are restricted to the robot's local surroundings and continuously guide a dynamics-aware trajectory planner. Similar to [15], we use the current knowledge of the global map and local observations as inputs to our policy, but also include the robot's dynamical state to allow for reasoning about the behavior of the underlying MPC.

### 2.1.3 Contribution

The main contributions of this work are the following:

- An informative trajectory planning hierarchical framework combining a viewpoint recommendation policy with receding-horizon trajectory optimization. Our method plans safe and dynamically feasible trajectories while navigating the robot to informative observations.
- A method for training a DRL agent together with a local motion planner, such that the policy learns to guide the motion planner in an obstacle-rich environment and to maximize the cumulative information-theoretic reward.

We present simulation results comparing our method with an MCTS planner and a greedy policy, in terms of the execution time and information gathering performance. We aim at significantly faster execution than MCTS, with little loss of performance, and substantially better performance than with the greedy policy. Additionally, we present qualitative results demonstrating the exploration behavior of our method.

## 2.2 Preliminaries

### 2.2.1 Problem Formulation

Consider a robot that has to explore an unknown 2D environment $\mathcal{W} \subset \mathbb{R}^2$ in order to find an unknown number of targets in this environment. The dynamics of the robot are described by a discrete-time model $\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t)$ where $\mathbf{x}_t$ is the state of the robot, and $\mathbf{u}_t$ is the control input applied at time step $t$. We assume that $\mathbf{x}_t$ is observed, e.g., using onboard sensing. We denote the position of the robot in $\mathcal{W}$ at time $t$ by $\mathbf{p}_t = [x_t, y_t]^T$, $\mathbf{p}_t \in \mathcal{W}$. The area covered by the robot at time $t$ is denoted by $\mathcal{O}_t$. The robot must avoid collisions with static obstacles $\mathcal{O}_{\text{obst}} \subset \mathcal{W}$.

When moving in the environment, the robot builds, from sensor observations, a map about possible target locations in the environment. We model this target map as a probabilistic occupancy grid map [52], represented by the random variable $\mathbf{M}$ (see Section 2.2.2).

The observation vector is modeled as a random variable $Z_t$, with a realization denoted by $z_t$. At each time step $t$ the robot makes an observation $Z_t$ about nearby targets at its current position $\mathbf{p}_t$, and updates its belief about the target map $\mathbf{M}$. Subsequently, the control inputs are computed that move the robot to its next observation pose $\mathbf{p}_{t+1}$.

The objective of the robot is to reduce the uncertainty in the target map $\mathbf{M}$ by making informative observations $Z_t$. We formalize this objective as maximizing the cumulative mutual information (MI) between the robot's prior about $\mathbf{M}$ at time step $t$, and the latest measurement $Z_t$, given the history of measurements until the last time step, $z_{0:t-1}$. The MI quantifies the reduction in uncertainty by making observation $Z_t$, and it is denoted by $I(\mathbf{M}; Z_t | z_{0:t-1})$ [53].

The informative trajectory planning problem then is to maximize the cumulative MI while ensuring a collision-free, kinodynamically feasible trajectory over a horizon $L$ (the total time-budget for the mission) and given an initial state $\mathbf{x}_0$ and an initial observation $z_0$:

$$
\begin{aligned}
\max_{\mathbf{u}_{0:L-1}} \quad & \sum_{t=1}^{L} I(\mathbf{M}; Z_t | z_{0:t-1}) && \text{(2.1a)}\\
\text{s.t.} \quad & \mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}) && \text{(2.1b)}\\
& \mathcal{O}_t \cap \mathcal{O}_{\text{obst}} = \emptyset && \text{(2.1c)}\\
& Z_t = h(\mathbf{x}_t), && \text{(2.1d)}\\
& \mathbf{x}_t \in \mathcal{X}, \mathbf{u}_{t-1} \in \mathcal{U}, && \text{(2.1e)}\\
& t = 1, \dots, L
\end{aligned}
$$

where (2.1b) is the constraint on the robot dynamical model (Section 2.2.3), (2.1c) is the collision avoidance constraints, and $\mathcal{X}$,$\mathcal{U}$ are the admissible sets of robot states and control inputs, respectively. Equation (2.1d) is the observation model, described in Section 2.2.2.

### 2.2.2 Belief Map and Observation Model

#### Belief Map

The target map $\mathbf{M}$ is a discretization of $\mathcal{W}$ in $n$ grid cells, associated with independent Bernoulli random variables $M_i \in \{0,1\}$, $\forall i \in \{1, \dots, n\}$, with 1 indicating target occupancy, and 0 otherwise. The robot's *belief* about the map $\mathbf{M}$ is described by probabilities of target occupancy in each cell $i$, denoted by $P_{t,i} \coloneqq \mathbb{P}(M_i = 1 | z_{0:t})$, and initialized with a uniform prior of $\mathbb{P}(M_i = 1) = 0.5$. Given a new observation $z_t$, the Bayesian update of $P_{t,i}$ using log odds [52] is:

$$
l(M_i | z_{0:t}) = l(M_i | z_{0:t-1}) + l(M_i | z_t), \tag{2.2}
$$

where $l(M_i | z_t)$ is an inverse sensor model [52].

#### Observation Model

To make observations $Z_t$, the robot is equipped with a sensor that can detect targets up to a distance $d_{\text{max}}$ from the robot and within a field-of-view of 360° and associate it with a cell in the map $\mathbf{M}$. The set of cells visible from position $\mathbf{p}_t$ is denoted by $\mathcal{I}_t$. It only includes cells for which the visibility of its center point is not occluded by obstacles $\mathcal{O}_{obst}$. The observation made at each time step is a vector composed of the individual cell observations of target occupancy, namely $Z_t \in \{0,1\}^{|\mathcal{I}_t|}$. Equation (2.2) is only applied to the cells in $\mathcal{I}_t$ after each observation. The mutual information between the prior about the target map $\mathbf{M}$ and an observation $Z_t$ is equal to the reduction of the conditional entropy in $\mathbf{M}$ by observation $Z_t$ [53, 54]:

$$I(\mathbf{M}; Z_t | z_{0:t-1}) = H(\mathbf{M}|z_{0:t-1}) - H(\mathbf{M}|Z_t, z_{0:t-1}) \tag{2.3}$$

where $H(\mathbf{M}) = \sum_{i=1}^{n} H(M_i)$ is the entropy of the target map and $H(M_i)$ are the respective cell entropies.

### 2.2.3 Robot Dynamics

We consider the robot to be modeled by a second-order unicycle model [55]

$$\begin{aligned} \dot{x} &= v\cos\psi & \dot{v} &= u_a & \dot{\psi} &= \omega \\ \dot{y} &= v\sin\psi & \dot{\omega} &= u_\alpha. \end{aligned} \tag{2.4}$$

which is discretized with sampling time $T_S$. Thus the state of the robot is described by $\mathbf{x} = [x, y, \psi, v, \omega]^T$, where $\psi$ the heading angle in a global frame, $v$ denotes the robot's longitudinal velocity, and $\omega = \dot{\psi}$ the angular velocity. The control input $\mathbf{u} = [u_a, u_\alpha]^T$ consists of the robot's linear and angular acceleration, respectively.

## 2.3 Method

We hierarchically solve the problem in (2.1) by separating it into a high-level sequential decision-making problem and a local trajectory planning problem. The first aims at determining a reference viewpoint, such that future information gains are maximized based on the current belief following from past observations (Section 2.3.1). The local trajectory planning problem aims at moving the robot towards the recommended viewpoint while ensuring kinodynamic feasibility and collision avoidance (Section 2.3.2). The concept of the proposed framework is depicted in Figure 2.1. Our proposed approach builds on [13], extending its task and environment scope for information gathering in obstacle-rich environments.

### 2.3.1 Reinforcement Learning of Viewpoint Recommendations

Our method learns, via reinforcement learning, a policy $\pi$ that recommends every $N_a$ timesteps a reference position $\mathbf{p}_t^{\text{ref}}$ in the robot's neighborhood (the reference viewpoint) to an MPC motion planner, such that the resulting trajectories of the robot lead to observations that maximize rewards, and eventually result in near-complete coverage of the available information in the environment.

#### Observation

The goal is to learn a policy that uses the robot's own belief about $\mathbf{M}$, and local observations about nearby obstacles $\mathbf{O}_t \in \mathcal{O}_{\text{obst}}$. Both inputs are visualized in Figure 2.2. The local obstacle observation $\mathbf{O}_t$ is a binary grid map of obstacles around the robot, given as an $m \times m$ image, centered at the robot's position and aligned with its orientation [15, 56].

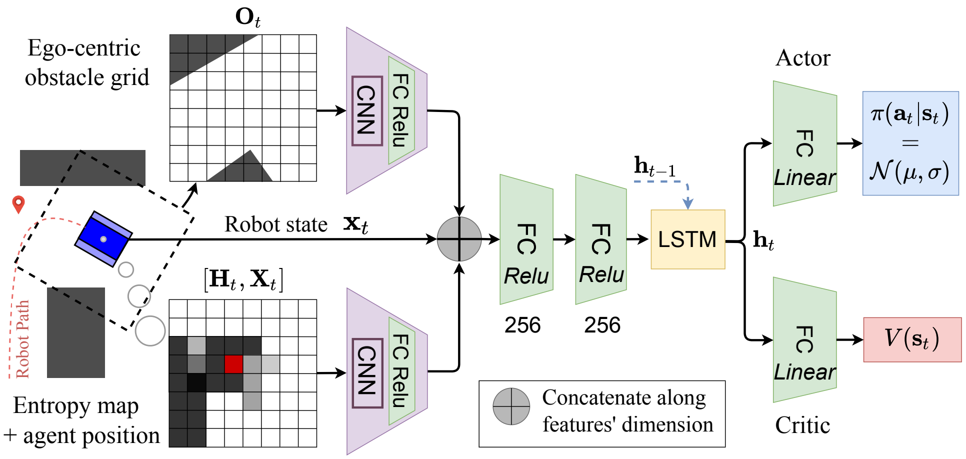

$\mathbf{O}_t$
Ego-centric obstacle grid
Robot state $\mathbf{x}_t$
$[\mathbf{H}_t, \mathbf{X}_t]$
Robot Path
Entropy map + agent position
CNN
FC Relu
CNN
FC Relu
FC Relu
256
FC Relu
256
$\mathbf{h}_{t-1}$
LSTM
$\mathbf{h}_t$
Actor
FC Linear
$\pi(\mathbf{a}_t|\mathbf{s}_t) = \mathcal{N}(\mu, \sigma)$
FC Linear
$V(\mathbf{s}_t)$
Critic
Concatenate along features' dimension

Figure 2.2: Proposed policy and value function network, with two encoders processing an ego-centric obstacle grid map $\mathbf{O}_t$ and a two-channel image representing the belief $[\mathbf{H}_t, \mathbf{X}_t]$, respectively. The entropy map $\mathbf{H}_t$ is depicted as a grayscale image, with darker shades corresponding to lower uncertainties. The second channel $\mathbf{X}_t$ is visualized by a red grid cell marking the current agent position. The encoder structure and hyperparameters are as in [56], $\mathbf{h}$ is the LSTM hidden state, and FC refers to a fully-connected layer.

Such an egocentric observation improves generalization across different environments. The robot's belief about $\mathbf{M}$ is represented by a map of cell entropies $H(M_i)$, denoted by $\mathbf{H}_t$, that informs the agent about uncertainties in different map regions. An indicator function map $\mathbf{X}_t$ for the agent's position in the map is appended as a second channel to $\mathbf{H}_t$ [27, 39]. Hence, at time step $t$ the RL observation vector $\mathbf{s}_t$ is

$$\mathbf{s}_t = [\mathbf{H}_t, \mathbf{X}_t, \mathbf{O}_t, \mathbf{x}_t]^{\mathrm{T}}, \tag{2.5}$$

where $\mathbf{x}_t$ is the robot's state defined in Section 2.2.3.

### Action

The RL action $\mathbf{a}_t \in \mathcal{A} \subset \mathbb{R}^2$ is defined as the relative position $\delta_t$ of the viewpoint reference with respect to the robot's current position,

$$\begin{aligned} \mathbf{a}_t &= \delta_t \sim \pi(\mathbf{a}_t|\mathbf{s}_t) \\ \mathbf{p}_t^{\mathrm{ref}} &= \mathbf{p}_t + \delta_t \end{aligned} \tag{2.6}$$

The position increment is constrained inside a square around the robot, such that the continuous action space of our RL method is $\mathcal{A} = \{\, \delta_t \in \mathbb{R}^2 \mid \|\delta_t\|_\infty \le \delta_{\max}\}$.

### Reward Function

The main objective of the informative trajectory planning problem (2.1) is to maximize the information gains. Hence, we define an information-theoretic reward function using the mutual information gained through the observation $Z_{t+N_a}$, made $N_a$ steps after the last action $\mathbf{a}_t$. Moreover, we add a term $r_{\mathrm{pen}}$ penalizing each time step, incentivizing the agent to achieve the coverage goal, terminating the episode, as soon as possible. The reward function is defined as

$$r(\mathbf{s}_t, \mathbf{a}_t) = I(\mathbf{M}; Z_{t+N_a} | z_{0:t}) + r_{\text{pen}}. \tag{2.7}$$

**Policy Network Architecture**

Figure 2.2 depicts our proposed policy network architecture. We employ two CNN models using the architecture and hyperparameters proposed in [56] to encode spatial information in the two image inputs $[\mathbf{H}_t, \mathbf{X}_t]$ and $\mathbf{O}_t$. These encoder networks are each trained by gradients coming from both the policy update loss (Section 2.3.3) and a reconstruction loss generated using a decoder network reversing the encoder operations [56]. Thus, compressed latent representations of spatial features in the local obstacle grid and the entropy map are learned, which the policy can exploit to learn actions that guide the robot around nearby obstacles and to map regions with high uncertainties. The two latent feature vectors are concatenated with the state $\mathbf{x}_t$ of the robot's dynamical model, so that the policy can learn how to guide the MPC planner using viewpoint references with respect to its closed-loop dynamical behavior.

After feeding the full feature vector into two fully-connected (FC) layers, an LSTM layer [57] models the time-dependencies between previous states and the current state. The hidden state of the LSTM is fed to the final actor and critic heads modeled as FC linear layers. We model the policy $\pi$ as a diagonal Gaussian distribution, i.e. $\pi_\theta(\mathbf{a}_t|\mathbf{s}_t) = \mathcal{N}(\mu, \sigma^2)$, such that $\delta_t \sim \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$. The distribution's mean $\mu$ and log-standard-deviation $\log \sigma$ are learned by the actor head. The critic head estimates the state-value function $V_\theta^\pi(\mathbf{s}_t) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t)\right]$ of the current policy, where $\gamma$ is the discount factor. The subscript $\theta$ denotes the current network parameters.

## 2.3.2 Local Planner

We rely on receding-horizon trajectory optimization to generate control commands for the robot satisfying dynamic and collision constraints. For dynamic constraints we employ a second-order unicycle model as defined in Section 2.2.3.

For the collision constraints, we assume the robot's space, $\mathcal{O}_t$, to be a circle with radius $r$, and each obstacle's space is defined as a polygon. To ensure collision-free motions, first, we compute a linear constraint to ensure that the robot's space does not overlap with static obstacle's space, i.e., $\mathcal{O}_t(\mathbf{x}_t) \cap \mathcal{O}_{\text{obst}} = \emptyset$, at planning step $k$ defined as

$$c_k^{o_j} = {\mathbf{n}_k^{o_j}}^{\mathrm{T}} \mathbf{p}_k \leq b_j - r, \tag{2.8}$$

where $\mathbf{n}_k^{o_j}$ is the normal vector at the closest point $\mathbf{p}_k^{o_j}$ on the surface of the $j$/th obstacle and $b_j = -{\mathbf{n}_k^{o_j}}^{\mathrm{T}} \mathbf{p}_k^{o_j}$. To limit the complexity of the optimization problem, we only consider a set of $n_{obs}$ constraints for the closest obstacles. The distance between the robot's position and the $j$-th linear constraint is computed as:

$$\left\| \mathbf{p}_k, c_k^{o_j} \right\| = \frac{\left| {\mathbf{n}_k^{o_j}}^{\mathrm{T}} \mathbf{p}_k + b_j \right|}{\left\| \mathbf{n}_k^{o_j} \right\|} \tag{2.9}$$

The DRL policy provides a reference position $\mathbf{p}_t^{\text{ref}}$ (viewpoint) guiding the robot to maximize future rewards. Similarly to [13], we define a terminal cost enabling the robot to reach the provided viewpoint reference:

$$J_N(\mathbf{x}_{t+N}, \mathbf{p}_t^{\text{ref}}) = \left\| \frac{\mathbf{p}_{t+N} - \mathbf{p}_t^{\text{ref}}}{\mathbf{p}_t - \mathbf{p}_t^{\text{ref}}} \right\|_{Q_N}, \tag{2.10}$$

where $\mathbf{p}_{t+N}$ is the robot's terminal position (at planning step $N$) and $Q_N = \text{diag}(q_N, q_N)$ is the terminal cost matrix. To generate smooth trajectories, we employ a quadratic penalty on the control commands as a stage cost for planning step $k$:

$$J_k^{\mathbf{u}}(\mathbf{u}_k) = \|\mathbf{u}_k\|_{Q_u} \tag{2.11}$$

where $Q_u = \text{diag}(q_a, q_\alpha)$ is the stage cost matrix.

At every time step $t$, a non-convex optimization problem is solved with planning horizon $N$ under the kinodynamic and collision constraints, given the initial state $\mathbf{x}_t \in \mathcal{X}$:

$$\begin{aligned} \min_{\mathbf{u}_{t:t+N-1}} \quad & \sum_{k=t}^{t+N-1} J_k^u(\mathbf{u}_k) + J_N(\mathbf{x}_{t+N}, \mathbf{p}_t^{\text{ref}}) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \\ & c_{k+1}^{o_j} \leq b_j - r, \ \forall j \in \{1, \dots, n_{\text{obs}}\}, \\ & \mathbf{u}_k \in \mathcal{U}, \ \mathbf{x}_{k+1} \in \mathcal{X}, \ \forall k \in \{t, \dots, t+N-1\}. \end{aligned} \tag{2.12}$$

The equality constraint is the discrete-time model of the continuous dynamics model presented in (2.4).

### 2.3.3 Training Procedure

First, we warm-start the policy training with behavior cloning updates, using the one-step greedy policy outlined in Section 2.4.2, which outputs $\mathbf{p}_t^{\text{ref}}$, in combination with MPC (2.12) as the expert policy. We define the expert reference viewpoint as $\mathbf{a}_t^* = \mathbf{p}_N^* - \mathbf{p}_t$, where $\mathbf{p}_N^*$ is the last position in the MPC-generated trajectory. For the first $N_{SL}$ policy steps of the training, we apply $\mathbf{a}_t^*$ as the agent's action and use it as a label to perform supervised learning of the policy $\pi_\theta$, as described in [13]. Subsequently, the policy is trained with DRL using Proximal Policy Optimization (PPO) [58] until reaching $N_{\text{train}}$ policy steps. One policy step yielding a new viewpoint corresponds to $N_a$ timesteps, and MPC is executed at every time step $t$ with the last sampled viewpoint reference. The PPO horizon is $S = L/N_a$ policy steps.

Training and testing are performed in randomly generated environments as depicted in Figure 2.3, with the agent initialized at a random position. Random rectangular obstacles are generated and environments, where obstacles block the agent from reaching the entire free space, are omitted. We employ curriculum learning during training [59], increasing the number of obstacles $N_{\text{obst}}$ from one to three.

Episodes are terminated after the completion of the information gathering task, or if a maximum number of time steps $t_{\max}$ is reached (*failure*). The task is completed when, at

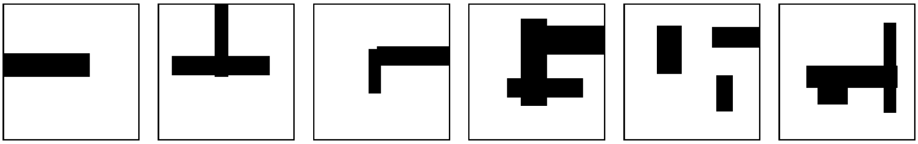


Figure 2.3: Examples of the random environments used in training.

time $t$, the conditional entropy of the belief about the map cells in the free space $\mathcal{W} \setminus \mathcal{O}_{\text{obst}}$, denoted by $\mathbf{M}^{\text{free}}$, is smaller than a predefined ratio of the entropy of the initial belief prior. That is, when $H(\mathbf{M}^{\text{free}}|z_{0:t}) \leq (1-\beta)H(\mathbf{M}^{\text{free}})$, where $\beta \in [0, 1)$ is the share of information that should be gathered by the robot, defining the coverage goal.

## 2.4 Results

In this section, we present quantitative and qualitative simulation results of the proposed method. We compare the performance metrics of our method against two baseline approaches (Section 2.4.3) and analyze the behavior of the informative trajectory planning method (Section 2.4.4). The baselines are introduced in Section 2.4.2, and the simulation setup for training and testing is outlined in Section 2.4.1.

### 2.4.1 Simulation Setup

The training procedure described in Section 2.3.3 builds upon the open-source PPO2 implementation provided by the Stable Baselines framework [60]. The nonlinear optimization problem (2.12) is solved using the Forces Pro solver [61]. Simulations are run in the environment provided by the open-source ”gym-collision-avoidance” package [62]. We train the policy $\pi_\theta$ with $N_{\text{proc}}$ processes for rollouts on a desktop computer with an AMD Ryzen 9 CPU and 64 GB of RAM. Table 2.1 presents the hyperparameters used.

### 2.4.2 Baselines

We compare the performance against two baseline approaches: one myopic and one non-myopic informative path planning method. Similar to our approach, we use both baselines to compute a reference viewpoint $\mathbf{p}_t^{\text{ref}}$ for the MPC.

**Myopic Greedy Viewpoint Selection**

As a myopic baseline, we use a one-step next-best-view planner similar to [46]. At each time step, we uniformly sample $N_{nbv} = 30$ points $\tilde{\mathbf{p}}_i, \forall i = 1, ..., N_{nbv}$ in the policy's action space $\mathcal{A}$, and evaluate the objective $I(\mathbf{M}; Z(\tilde{\mathbf{p}}_i)|z_{0:t})$ for expected observations $Z(\tilde{\mathbf{p}}_i)$ at these viewpoints. The point with the highest reward is chosen and passed as $\mathbf{p}_t^{\text{ref}}$ to the MPC. This greedy method is also used for warm-starting the training, as explained in Section 2.3.3.

**Non-Myopic Monte Carlo Tree Search (MCTS)**

We use an MCTS planner [30–32] as a baseline to find finite-horizon sequences of viewpoints that maximize cumulative information rewards. We build on an open-source Python implementation of Dec-MCTS [31], and use it for single-robot planning. The planner uses

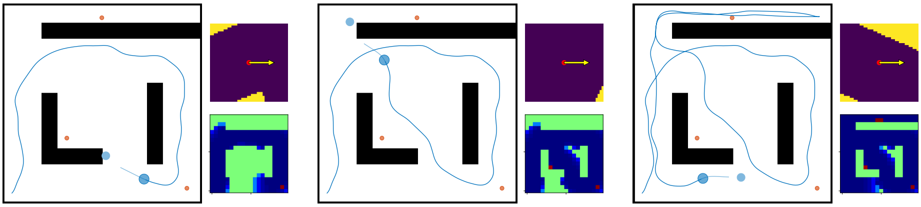
Table 2.1: Hyperparameters.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **MPC** | | | | | | | |
| Horizon $N$ | 15 | $T_S$ | 0.1 s | $q_N$ | 5.0 | $q_a$ | 0.003 |
| $q_\alpha$ | 0.003 | – | – | – | – | – | – |
| **Training** | | | | | | | |
| Learning rate | $10^{-4}$ | Horizon $S$ | 128 | Clip range | 0.2 | $\gamma$ | 0.99 |
| $N_{\text{train}}$ | $2 \cdot 10^7$ | $N_{\text{epochs}}$ | 2 | $N_{\text{proc}}$ | 16 | $N_a$ | 5 |
| $\delta_{\max}$ [m] | 4 | $t_{max}$ | 640 | $r_{pen}$ | -0.1 | $N_{SL}$ | $10^6$ |
| **MCTS Baseline** [31] | | | | | | | |
| $N_{\text{tree}}$ | 100 | $N_{\text{sim}}$ | 10 | $H_{\text{MCTS}}$ | 4 | $T_{mp}$ [s] | 1.2 |
| $u_v$ [m/s] | $\{0, 1, 3\}$ | $C_{\text{UCB}}$ | 2.0 | $u_\omega$ [rad/s] | $\{-\pi/4, -\pi/10, 0, \pi/10, \pi/4\}$ | | |

Table 2.2: Performance results, aggregated over 100 random maps with $n \in \{1, 2, 3\}$ obstacles.

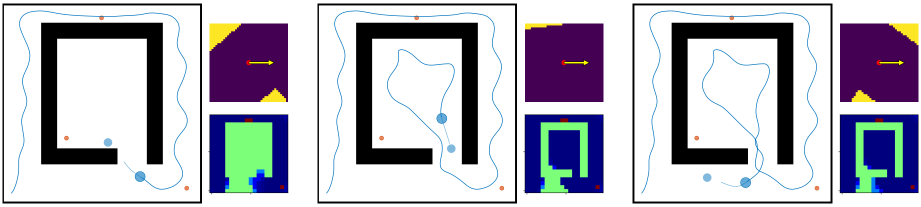
| **Metric** | **MCTS** | **Greedy** | **Viewpoint Policy (ours)** |
|---|---|---|---|
| Avg. episode rewards (mean ± std) | | | |
| $N_{\text{obst}} = 1$ | 19.60 ± 1.99 | 18.98 ± 2.25 | 19.41 ± 0.89 |
| $N_{\text{obst}} = 2$ | 18.24 ± 1.09 | 17.50 ± 2.65 | 18.03 ± 1.17 |
| $N_{\text{obst}} = 3$ | 16.79 ± 2.00 | 15.64 ± 3.75 | 16.49 ± 1.41 |
| Failed episodes [%] | | | |
| $N_{\text{obst}} = 1$ | 1 | 6 | 0 |
| $N_{\text{obst}} = 2$ | 2 | 2 | 1 |
| $N_{\text{obst}} = 3$ | 2 | 8 | 1 |
| Time until completion [s] | | | |
| $N_{\text{obst}} = 1$ | 46.8 | 56.7 | 53.6 |
| $N_{\text{obst}} = 2$ | 50.7 | 61.6 | 55.8 |
| $N_{\text{obst}} = 3$ | 51.7 | 65.7 | 59.6 |
| Avg. Runtime [s] | 2.486 | 0.046 | 0.004 |

a simplified first-order kinematic model of the robot dynamics and a small set of motion primitives as discrete action space. The motion primitives are combinations from different velocity and angular velocity inputs, $u_v$ and $u_\omega$, as given in Table 2.1, with a length of $N_a$ timesteps. The planning horizon is $H_{\text{MCTS}}$, for each replanning, $N_{\text{tree}}$ MCTS iterations are performed, and each new leaf node is evaluated with $N_{\text{sim}}$ rollouts. The first position in the best plan is passed as $\mathbf{p}_t^{\text{ref}}$ to the MPC. We replan at every time step $t$, but the planning time does not affect the simulated time due to the sequential implementation. Thus the robot does not have to stop for replanning. Furthermore, our MCTS implementation has access to the global obstacle map for computing rewards of possible future positions.



(a) $t = 22.5\,\mathrm{s}$ (b) $t = 30\,\mathrm{s}$ (c) $t = 71\,\mathrm{s}$

Figure 2.4: Trained policy behavior in an *unstructured* environment of higher complexity than in training, with three timesteps of an episode displayed. The agent effectively explores all areas of the environment and manages to enter and leave the narrow dead-end corridor. The upper-right grid next to each map shows the ego-centric local obstacle observation $\mathbf{O}_t$ of the agent, and the lower-right grid the belief map of the probabilities $P_{t,i}$ of the current belief (Section 2.2.2). The colors in the belief map range from dark blue $P_{t,i} = 0$ to dark red $P_{t,i} = 1$, with the green areas indicating $P_{t,i} = 0.5$ (the initial value).



(a) $t = 30.5\,\mathrm{s}$ (b) $t = 48\,\mathrm{s}$ (c) $t = 53\,\mathrm{s}$

Figure 2.5: Trained policy behavior in a *structured* environment of higher complexity than in training, with three timesteps of an episode displayed. The three snapshots show that the robot is effectively guided into and out of a room-like structure, without visiting areas twice. The setup of the figures is as described in Figure 2.4.

### 2.4.3 Performance Results

This section presents the quantitative performance results of our method and the two baselines. The results, summarized in Table 2.2, are aggregated over a set of random environments for three map complexity levels defined by the number of sampled obstacles. For each number of obstacles, 100 random scenarios are simulated for each of the methods. In each episode, the agent has a maximum of $t_{\text{max}}$ to reach the coverage goal of $\beta = 0.9$ before it is considered a *failure*. We quantify the performance by the average cumulated reward over an episode, the percentage of failure episodes, the average travel time, and the average runtime (excluding the MPC) of the three viewpoint recommending methods.

Our method outperforms the greedy next-best-view baseline in terms of average episode rewards, completion time, and failures for all map complexities. The greedy method exhibits a large number of failures because it cannot reason about unexplored areas outside the local surroundings. Thus the robot often revisits already explored areas multiple times instead of moving to unexplored areas to complete coverage. Moreover, our method achieves the lowest percentages of failure episodes and the lowest execution times. Failures of the MCTS planner occur when it determines viewpoint references that are unreachable for the MPC, which our method avoids by training with the MPC. The long runtimes of MCTS are caused by the expensive computation of the set of visible cells $\mathcal{I}_t$ (Section 2.2.2) for a large number of viewpoint candidates during planning, necessary to evaluate their information gain. In contrast, our trained policy $\pi$ can infer a promising viewpoint reference only from currently available observations. This comes at the cost of suboptimal average rewards and completion time compared to the MCTS planner. Note, however, the MCTS planner's advantageous assumptions (Section 2.4.2), as the long runtime does not affect performance and the global obstacle knowledge enable evaluating rewards for distant positions during planning.

### 2.4.4 Qualitative Analysis

This section analyzes the behavior of our proposed method in two scenarios not used during training and with higher complexity than the training scenarios, in terms of obstacle placement and an increased coverage goal of $\beta = 0.95$. Figures 2.4 and 2.5 show the agent path for three different time steps with the recommended viewpoint, the local observation, and belief map of the agent for each scenario. In Figures 2.4a to 2.4c, the viewpoint reference leads the agent into the most promising unobserved areas and enables it to enter and leave a narrow dead-end corridor at the top of the map. While not globally optimal, the behavior exhibits an efficient strategy of guiding the robot towards unobserved areas, maximizing information gains, and dealing with difficult environment structures. In Figures 2.5a to 2.5c, the robot is able to enter and leave a room-like structure. The policy guides the robot to observe inside the room when reaching the entrance, instead of moving further, and decides to leave the room as soon as almost all available information has been gathered. Subsequently, it guides the robot into the remaining unobserved areas.

## 2.5 Conclusions and Future Work

In this paper, we introduced a navigation policy capable of guiding a local trajectory planner towards maximizing the information gathered in an unknown environment. We employed reinforcement learning to learn the information-gathering policy using only locally available observations and previously gathered information. The policy learns to maximize information-theoretic rewards by providing a viewpoint reference that an MPC-based local motion planner uses to generate trajectories respecting the robot's safety constraints. The results show that the learned policy is able to effectively guide the robot through unseen environments, and achieve quantitative performance comparable to an MCTS planner. Moreover, our method can be run at a rate three orders of magnitude faster than the MCTS planner, allowing for quick reactions in dynamic scenarios. Future work will consider a limited field of view and experiments on a real robotic platform.

# 3

# Learning Semantic Priorities for Autonomous Target Search

*The use of semantic features can improve the efficiency of target search in unknown environments for robotic search and rescue missions. Current target search methods rely on training with large datasets of similar domains, which limits the adaptability to diverse environments. However, human experts possess high-level knowledge about semantic relationships necessary to effectively guide a robot during target search missions in diverse and previously unseen environments. In this chapter, we propose a target search method that leverages expert input to train a model of semantic priorities. By employing the learned priorities in a frontier exploration planner using combinatorial optimization, our approach achieves efficient target search driven by semantic features while ensuring robustness and complete coverage. The proposed semantic priority model is trained with several synthetic datasets of simulated expert guidance for target search. Simulation tests in previously unseen environments show that our method consistently achieves faster target recovery than a coverage-driven exploration planner.*

This chapter is based on: ***M. Lodel***, *N. Wilde, R. Babuska, J. Alonso-Mora, "Learning Semantic Priorities for Robotic Target Search Planning", under review.*

## 3.1 Introduction

Autonomous robots that can explore unknown environments efficiently by searching for objects of interest (OOI) are promising tools in applications such as search and rescue, inspection, and environmental monitoring. Efficient search typically relies on reasoning about semantic information in the scene and consequently determining *where to search* next. For example, search and rescue in an industrial site likely focuses on zones frequently used by workers, such as offices and storage rooms, that can be identified by characteristic objects like desks or shelfs.

By leveraging semantic priors of typical object arrangement, recent works [6–8, 16, 19, 20, 23, 63] have demonstrated this semantic exploration paradigm and achieved effective autonomous search behavior. However, these methods either train on large domain-specific datasets [64, 65] or use foundation models trained on internet-scale datasets, leading to common-sense reasoning capabilities [7, 23, 63].

However, domain-specific training data may not always be available for highly unpredictable and specific environments. Moreover, foundation models require extensive computational resources that are infeasible for onboard deployment. On the contrary, pure coverage exploration methods [18, 35, 66] that effectively search everywhere can be deployed independently of domain priors but can take a long time to find OOIs.

Specialized human operators, such as first responders in search and rescue, often have high-level knowledge about promising search locations based on observed semantic features, such as relevant objects. However, increasing autonomy in exploration can be prefereable over teleoperation, as it reduces the operator's workload and is less reliant on robust communication. Hence, we aim to leverage expert inputs to learn semantic priors for autonomous target search. Independent of semantic information, coverage-driven exploration methods [18, 35, 66] can guarantee target discovery. Therefore, a reliable semantic search approach must ensure that efficient exploration of the entire environment continues independently of semantic features and their learned priorities.

In this paper, we present a hierarchical exploration framework (Figure 3.1), that can learn semantics target search strategies from expert inputs. Our paper makes the following contributions:

- We introduce a framework for learning a semantic priority function that models the knowledge driving expert interventions, instead of imitating the expert.
- We present a novel exploration planner leveraging these priorities to prioritize promising frontiers.

In simulation experiments, we show that our framework achieves more efficient target search than coverage exploration after learning from only a small set of expert interventions. Moreover, our approach exhibits robust target search performance when learning from different simulated expert behaviors.

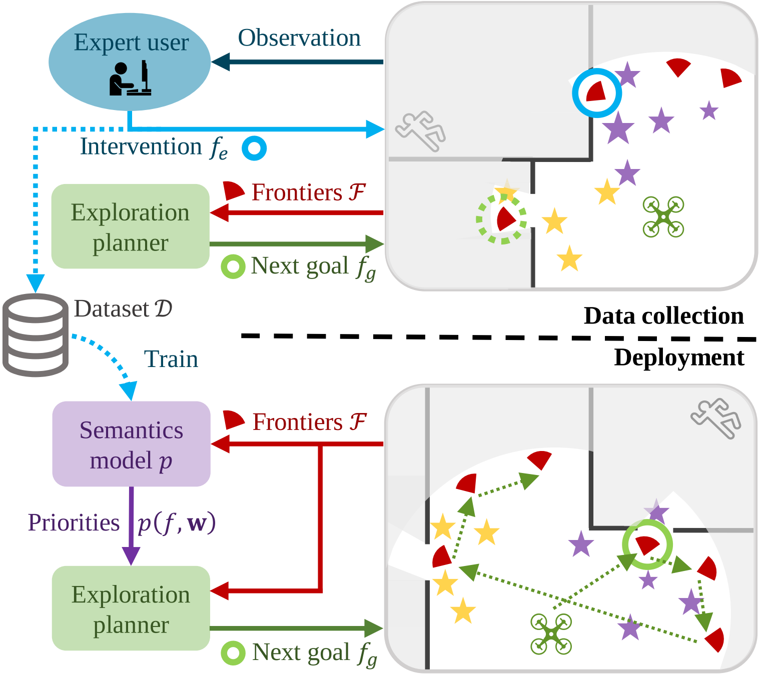


Figure 3.1: Conceptual overview of the proposed framework. During data collection, an expert generates interventions into the planner's goal output, prioritizing certain semantically relevant objects (depicted by stars). These are used to train a semantic model, which outputs priorities for each exploration frontier that, in turn, guide the exploration planner. The exploration planner outputs the next frontier viewpoint to navigate to.

## 3.2 Related Works

In this section, we discuss existing approaches and how they relate to our work, focussing on semantic target search, learning human objective functions and coverage exploration.

### Semantically-informed Target Search

Semantically informed target search exploits environmental semantic features to accelerate target localization. Several works address object search in unknown environments by learning semantic object relations from large-scale datasets [64, 65]. Reinforcement learning (RL) approaches [6, 16] train target search policies directly in simulation, whereas other methods [19, 20] predict the cost-to-go of different positions, demonstrating better data efficiency than RL. Conversely, zero-shot object search [7, 23, 63] show that foundation models trained on internet-scale data can be used to predict likely object locations from semantic context in common indoor environments. The authors of [8] distill semantic knowledge from a large language model (LLM) into a smaller model for online inference of target probabilities. In our paper, we learn a model of semantic priorities, similar to the prediction approaches [19, 20]. We learn semantic knowledge from expert inputs unlike prior work using environment data [6, 16, 19, 20], comparable to distilling LLM common sense reason in [8].

### Learning Human Objectives

Learning a priority model from human feedback involves learning the human's objective function. In most works, this is formalized as learning a reward function [24, 25, 67] or action-value function [68, 69]. *Offline* feedback methods [24, 25, 68] query the human for choice of different precomputed system behaviors [24, 25] or with states requiring a goal demonstration [68]. However, generating such queries is challenging in uncertain long-horizon tasks like exploration. With *online* feedback [67, 69], the human chooses when to provide inputs as he interacts with an agent executing some baseline behavior. Such online inputs can be binary feedback [67] or interventions with low-level demonstrations [69]. Our method considers online feedback in the form of expert interventions, similar to [69], demonstrating the preferred exploration frontiers. We propose to learn an exploration priority model of different frontiers, similar to learning a value function over planning goals [68]. Moreover, we employ a stochastic model of expert actions, as in [25].

### Coverage-driven exploration

Coverage-driven exploration methods maximize the expected area coverage in order to build an occupancy map without considering semantic features. Recently proposed methods employ combinatorial planning to visit all exploration frontiers [18, 35, 66, 70], or navigation policies trained to maximize future coverage rewards using RL [27, 71]. Combinatorial planners repeatedly compute tours over all frontiers, allowing reasoning over long horizons and efficient navigation across frontiers. These approaches have proven to work robustly in challenging real-world experiments [18, 35, 66, 70]. We build on this concept and employ a combinatorial planner over frontiers, but consider semantic features for target search in the planner. To this end, we propose a planner formulation that, similar to [18], can schedule frontiers based on a priority measure but prioritizes based on both semantics and coverage.

## 3.3 Problem Formulation

We consider the usecase where an autonomous robot searches for a target object in an unknown environment $\mathcal{W} \subset \mathbb{R}^2$ with obstacle-free space $\mathcal{W}_f \subset \mathcal{W}$. The robot's position at time $t$ is denoted by $\mathbf{x}_t \in \mathcal{W}_f$, and it starts exploring from an initial position $\mathbf{x}_0$. The robot moves incrementally with actions $\mathbf{a} \in \mathbb{R}^2$ bounded by $|\mathbf{a}| < \delta_{\max}$, where $\delta_{\max}$ is the maximum distance per time step. An action $\mathbf{a}$ can only be applied if the new position is in free space, i.e., $\mathbf{x}_t + \mathbf{a} \in \mathcal{W}_f$. A more complex dynamic model can be considered, for example, by tracking the reference $\mathbf{a}$ with a model predictive controller, as in [71].

### Occupancy Map

From range observations with sensing range $r$ until time $t$, the robot builds an occupancy map $\mathbf{M}_t$ of the environment. The occupancy map is represented as a grid, where cells correspond to evenly spaced positions $\mathbf{x} \in \mathcal{M}$, $\mathcal{M} \subset \mathcal{W}$, and are in one of three discrete states: unexplored (0), free (1), or occupied by obstacles (-1), i.e., $\mathbf{M} \in \{-1, 0, 1\}^{m \times m}$ with grid size $m$.

### Semantic Features

The robot observes objects of different semantic classes $\mathcal{S}$ when exploring the environment. An object is denoted as $o = (o^p, o^s)$, defined by its position $o^p \in \mathcal{W}_f$ and its semantic class $o^s \in \mathcal{S}$. We assume that the robot's sensors can detect objects around the robot within radius $r$ that are not occluded by obstacles. The set $\mathcal{O}_t$ denotes the objects observed up to time t. We further assume that semantic relationships between objects of different classes exist, i.e., the presence of certain objects can indicate an increased or reduced likelihood of other objects being present close by.

### Expert Input

We further assume the availability of an expert with knowledge of semantic relationships relevant to the search task. Leveraging this knowledge, the expert can infer likely target locations from the observed objects $\mathcal{O}_t$, and guide the robot to the target with waypoint inputs $\mathbf{h} \in \mathcal{W}_f$ that follow some expert policy $\mu$, i.e., $\mathbf{h}_t \sim \mu(\mathbf{h}_t|\mathcal{O}_t)$. From this expert interaction, a dataset $\mathcal{D}$ of expert inputs $\mathbf{h}_t$ with associated observations $\mathbf{M}_t, \mathcal{O}_t$ is recorded.

### Problem Statement

Given a dataset of human-guided target search trajectories $\mathcal{D}$, the problem is to find a navigation policy $\pi_{\mathcal{D}}$ controlling the robot with $\mathbf{a}_t = \pi_{\mathcal{D}}(\mathbf{M}_t, \mathcal{O}_t)$, that minimizes the distance traveled until discovering a target object $o_g$, using the map and object memory as current knowledge about the environment. With $H$ as the final time step, the problem is formulated as

$$\begin{aligned} \pi_{\mathcal{D}} = \arg\min_{\pi} & \sum_{t=0}^{H-1} \|\mathbf{x}_{t+1} - \mathbf{x}_t\| \\ \text{s.t.} \quad & \|o_g^p - \mathbf{x}_H\| \leq r \\ & \mathbf{x}_t = \mathbf{x}_{t-1} + \pi(\mathbf{M}_t, \mathcal{O}_t), \ \forall t \in \{1, \ldots, H\}, \end{aligned} \tag{3.1}$$

where the first constraint indicates target discovery at time $H$.

## 3.4 Method

Exploring an unknown environment to search for a target object requires continually solving two subproblems: Semantic scene understanding, or *where is it promising to explore*, and planning, or *where to go next*, given a set of regions to explore. We choose to solve these two problems in a hierarchical framework depicted in Figure 3.1 to obtain a data-efficient approach robust to unseen scenarios.

Both subproblems are solved using the concept of frontier exploration [72], which we formalize for our method in Section 3.4.1. The first problem of semantic scene understanding is formalized as evaluating different frontiers with a semantic priority function. Specifically, we present an approach to learning such a semantic priority function from expert interventions in Section 3.4.2. To solve the second problem of efficient navigation, we devise a combinatorial target search planner leveraging the learned semantic priority function. Specifically, the planner determines a visitation order such that semantically promising frontiers with increased probability of target discovery are prioritized, thus approximately solving Problem (3.1).

### 3.4.1 Frontier Exploration

In this section, we describe how our approach formalizes the concept of frontier exploration, drawing inspiration from recent works [18, 66]. Frontiers are the boundaries between explored and unexplored space in $\mathbf{M}_t$ and are used to derive a discrete set of candidate positions for observing unexplored space, called *frontier viewpoints*, that enable efficient exploration planning. To obtain such frontier viewpoints $f \in \mathcal{F}_t$, $\mathcal{F}_t \subset \mathcal{W}_f$ and efficient paths between them, a topological graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ is gradually constructed in the free space of $\mathbf{M}_t$. At every timestep, the graph is expanded using a sampling-based method from [18], ensuring sparsity. We consider every node $v_i \in \mathcal{V}_t$ as a potential frontier viewpoint if sufficient unexplored area is visible from $v_i$. To this end, we define a *coverage gain* function $\mathcal{I}(v_i) : \mathcal{V} \mapsto \mathbb{R}$ that denotes the gain in map coverage when observing frontiers from $v_i$. Specifically, the coverage gain approximates the expected gain in the covered area by casting a fixed number of equally spaced rays from $v_i$ and averaging the number of visible unexplored cells on each ray. The set of frontier viewpoint nodes $\mathcal{F}_t$ are those with $\mathcal{I}(v_i) > \mathcal{I}_{\text{thres}}$, referred to as *frontiers* $f \in \mathcal{F}_t$. We further assume that $\mathbf{M}_t$ is clustered into regions, e.g., rooms in a building, using a method such as [73], and each frontier is associated with a region.

### 3.4.2 Modeling Expert Frontier Choices

This section formulates a model of expert behavior that will be used to train the semantic priority model. When collecting data, the expert can *intervene* in the robot's exploration behavior at any time $t$ by determining the next *waypoint* that the robot will navigate to.

#### Semantic Priority Function

The expert considers each available frontier $f \in \mathcal{F}_t$ as a potential intervention waypoint, and evaluates how likely exploring a frontier $f$ leads to the target object, based on nearby objects and the expert's semantic knowledge. This evaluation is formalized as a semantic priority function $p(f, \mathbf{w})$. We model this function with a weighted sum where $\mathbf{w} \in [0, 1]^n$ are weights on $n$ different features, such as semantic classes. These features form a semantic feature vector $\phi(f)$ for each frontier $f$. Thus, the priority function can be written as,

$$p(f, \mathbf{w}) = \mathbf{w}^T \phi(f) \tag{3.2}$$

which is common in preference learning [24, 74] to allow learning from a small number of expert interventions. The weight vector $\mathbf{w}$ used by the expert is unknown and will be estimated from expert inputs.

#### Semantic Feature Vector

The feature vector $\phi(f)$ consists of two parts: Semantic features $\phi_s$ and an auxiliary region novelty feature $\phi_n$, i.e., $\phi(f) = [\phi_s(f),\ \phi_n(f)]^T$. Semantic features $\phi_s$ describe the occurrence of different semantic classes around the frontier node $f$. Each semantic feature needs to capture the presence of the semantic class in the vicinity and in the region of the frontier. Both effects are part of the semantic feature vector: A binary *local* semantic vector $\phi_{s,l}(f) \in \{0, 1\}^{|\mathcal{S}|}$ indicating if a class is visible within a small radius around $f$, and a binary *region* semantic vector $\phi_{s,r}(f) \in \{0, 1\}^{|\mathcal{S}|}$ indicating if a class is present in the same region as $f$. We combine both as $\phi_s(f) = \lambda\phi_{s,r}(f) + (1-\lambda)\phi_{s,l}(f)$ with $\lambda$ as hyperparameter. The region novelty feature $\phi_n$ captures the expert's interest in observing semantic information in unexplored regions, and remains 1 unless a small number of objects are observed in the region of frontier $f$.

**Expert Intervention Model**

When providing online waypoint interventions, the expert's capability to quickly plan over multiple frontiers is limited. Hence, we model the expert behavior with a greedy algorithm for choosing the next frontier. This greedy choice is modeled by a utility function $u(f)$ assigned to each frontier $f \in \mathcal{F}_t$. The expert is also interested in coverage exploration to guarantee search success without relying only on semantic priorities. Furthermore, the expert aims at minimizing the traveled distance until discovering the target object (Problem (3.1)), which is modeled by discounting the semantic priority by the traveling costs to the frontier. We propose a greedy choice model, maximizing a utility $u(f, \mathbf{w})$, that combines the semantic priority $p(f, w)$ with the coverage gain $\mathcal{I}(f)$ and the distance to the frontier, given by

$$u(f, \mathbf{w}, w_{\mathcal{I}}) = \delta(f)\big(p(f, \mathbf{w}) + w_{\mathcal{I}}\mathcal{I}(f)\big). \tag{3.3}$$

Here $\delta(f)$ is the distance-based discounting function, defined as

$$\delta(f) = 1 - \left({d_t(f)}/{\max_{f' \in \mathcal{F}_t} d_t(f')}\right) + \epsilon \tag{3.4}$$

with $d_t(f)$ expressing the traveling distance from the current position $\mathbf{x}_t$ to $f$ through $\mathcal{G}_t$ and $\epsilon$ defining the minimum discounting factor. The utility model in Equation (3.3) adds a coverage term weighted by the learnable parameter $w_{\mathcal{I}}$ to the semantic priority $p$ and discounts this extended priority by a factor $\delta(f)$ decreasing with distance to the frontier. Normalizing distances in $\delta(f)$ ensures consistent utility values across different frontier sets $\mathcal{F}$. Finally, the utility function can be written as a linear model $u(f, \mathbf{w}, w_{\mathcal{I}}) = u(f, \widetilde{\mathbf{w}}) = \widetilde{\mathbf{w}}^T\widetilde{\phi}(f)$ with augmented weights $\widetilde{\mathbf{w}} = [\mathbf{w}, w_{\mathcal{I}}]^T$ and features $\widetilde{\phi}(f) = \delta(f)\big[\phi(f),\ \mathcal{I}(f)\big]^T$.

**Pairwise Choice Model**

Next, we derive a probabilistic model of the expert's frontier choice to learn the expert weights from noisy expert intervention data. We model the expert preference for a frontier $f_e \in \mathcal{F}_t$ as pairwise choices between $f_e$ and all other available frontiers. Hence, the expert prefers frontier $f_e$ if its utility is higher than of all other available frontiers, i.e., if $u(f_e, \widetilde{\mathbf{w}}) \geq u(f, \widetilde{\mathbf{w}}),\ \forall f \in \mathcal{F}_t \setminus \{f_e\}$. The Bradley-Terry model [25, 75] defines the probability of choosing $f_i$ over $f_j$, denoted by $\mathbb{P}(f_i \succ f_j)$, as a logistic sigmoid function $\sigma$ of their utility difference, i.e., $\mathbb{P}(f_i \succ f_j) = \sigma(\beta(u(f_i) - u(f_j)))$. Here, $\beta$ is the rationality parameter modeling uncertainty in the expert's decision-making process. However, this model assumes that probabilities converge to 0 or 1 for large utility differences. We choose to modify this model to account for a residual error probability independent of the utility difference and $\beta$, considering cases where the utility model cannot capture potentially complex expert reasoning. Inspired by [24], we define $\rho \in [0, 0.5]$ as a lower bound on the probability of wrong choice independent of the utilities, used to formulate a scaled and shifted sigmoid function $\sigma_\rho$:

$$\sigma_\rho(x) = (1 - 2\rho)\sigma(x) + \rho. \tag{3.5}$$

Then, the probability that the expert chooses $f_e$ over any $f \in \mathcal{F}_t \setminus \{f_e\}$, given weights $\widetilde{\mathbf{w}}$, is modeled as

$$\mathbb{P}(f_e \succ f|\widetilde{\mathbf{w}}) = \sigma_\rho\big(\beta\, \widetilde{\mathbf{w}}^T(\widetilde{\phi}(f_e) - \widetilde{\phi}(f))\big). \tag{3.6}$$

Here, $\beta$ and $\rho$ are tunable hyperparameters. This proposed model captures noisy expert waypoint interventions based on the semantic priority function $p(f, \mathbf{w})$.

### Learning Expert Weights

The final step of the expert model is learning the expert weights from recorded intervention data. Given a set of $N$ choices $\mathcal{C} = \{(f_e^1, f^1), \ldots, (f_e^N, f^N)\}$ from the expert and assuming a uniform prior, we obtain the maximum likelihood estimate of the expert weights given the expert choices using gradient-based optimization, solving

$$\widetilde{\mathbf{w}}_{mle} = \underset{\widetilde{\mathbf{w}}}{\operatorname{argmin}} \sum_{(f_e, f) \in \mathcal{C}} \big[-\log \mathbb{P}(f_e \succ f|\widetilde{\mathbf{w}})\big], \tag{3.7}$$

## 3.4.3 Frontier Planning for Priority-Aware Exploration

In this section, we introduce a global planning method for target search given a semantic priority model (Section 3.4.2).

### Target Search as Combinatorial Optimization

We extend coverage-maximizing exploration methods that leverage combinatorial planning over frontier viewpoints [18, 35, 66, 70], by incorporating semantic priorities. The combinatorial planner generates a visitation order, or *tour*, through all known frontier viewpoints. For effective target search, promising frontiers should be scheduled earlier in the tour, such that the distance to the target object is minimized (Equation (3.1)). Consequently, we need to minimize the total distance traveled to frontiers with high semantic priority values $p(f, \mathbf{w})$, which are expected to be close to the target. We frame target search as a variant of the Minimum Latency Problem (MLP) [76], denoted as weighted MLP (WMLP), where the planned visitation latencies of the frontiers are weighted using the learned semantic priority model $p(f, \mathbf{w})$.

### Planner Formulation

We formulate the planning problem over a subset of nodes in the topological graph $\mathcal{G}_t$ composed of the the frontier nodes $\mathcal{F}_t$ and the robot's current node $v_t \in \mathcal{V}_t$, denoted as $\mathcal{F}'_t = \mathcal{F}_t \cup \{v_t\}$. A distance matrix $D$ contains the lengths of the shortest paths through $\mathcal{G}_t$ between all pairs of nodes in $\mathcal{F}'_t$. The tour $T$ is a sequence of all nodes in $\mathcal{F}'_t$ describing the planned visitation order, always starting with the robot node $v_t$. We denote that frontier node $f_i$ is scheduled at position $j$ in the tour as $T(j) = f_i$ for $j > 0$, while $T(0) = v_t$. Let $P(f)$ be a priority function that assigns each node in $\mathcal{F}'_t$ a priority weight, and $m = |\mathcal{F}'_t|$, then the WMLP objective is

$$\min_T \sum_{i=1}^{m-1} P(T(i)) \sum_{j=1}^{i} D(T(j-1), T(j)). \tag{3.8}$$

**Algorithm 1:** Prioritized exploration planning

```
Input: Semantic priority model weights $\mathbf{w}_{mle}$
Init $\mathcal{G}_t \longleftarrow \emptyset$, $\mathcal{F}_t \longleftarrow \emptyset$, and unexplored map $\mathbf{M}$
foreach time step t from 1 until $t_{\text{end}}$ do
    $\mathbf{M}_t, \mathcal{G}_t, \mathcal{F}_t, v_t \longleftarrow$ PerceptionUpdate()
    if Target found or $\mathcal{F}_t = \emptyset$ then
        break
    if $\mathcal{F}_t \neq \mathcal{F}_{t-1}$ or $\mathcal{I}(f)$ changed for any $f \in \mathcal{F}_t$ then
        $\mathbf{P} \longleftarrow$ FrontierPriorities($\mathcal{F}_t, \mathbf{w}_{mle}$)    ▷ Computes vector with Equation (3.9) $\forall f \in \mathcal{F}_t$
        $T \longleftarrow$ LNSsolver ($\mathcal{G}_t, \mathcal{F}_t, v_t, \mathbf{P}$)
        $f_g \longleftarrow T(1)$    ▷ Set goal node to next in tour
    else
        if $v_t = f_g$ then
            $f_g \longleftarrow$ next frontier in $T$
    $\mathcal{P} \longleftarrow$ ShortestPath($\mathcal{G}_t, v_t, f_g$)
    Move to next vertex in $\mathcal{P}$
```

Assuming a unit velocity, this problem minimizes a priority-weighted sum of the visitation latencies of each frontier, favoring earlier visits to high-priority frontiers. The priority function $P(f)$ leverages the learned semantic priorities $p(f, \mathbf{w}_{mle})$ to prioritize regions that likely lead to the target. Combining semantic priorities with expected coverage gain ensures robust exploration when the semantic priorities are ambiguous or incorrect, e.g., when encountering unseen states. Instead of the weighted sum model used in Equation (3.3), we propose a heuristic priority function $P(f)$ that always pursues coverage but is biased to semantically important frontiers, which we found more robust for the WMLP planner. Let $p'(f) = p(f)/p_{max,t}$ be the normalized semantic priority of frontier $f$ with $p_{max} = \max_{f \in \mathcal{F}_t} p(f)$, then $P(f)$ is given by

$$P(f) = \big(p'(f, \mathbf{w}_{mle}) + \alpha\big) \cdot \mathcal{I}(f). \tag{3.9}$$

Here, $\alpha \in [0, 1]$ is a hyperparameter controlling the trade-off between semantic priority and coverage gain. Note that while we learn the weight vector $\widetilde{\mathbf{w}}_{mle} = [\mathbf{w}_{mle}, w_{\mathcal{I},mle}]^T$, we only use $\mathbf{w}_{mle}$ for inferring frontier priorities, and discard the learned weight $w_{\mathcal{I},mle}$ of the coverage gain. This allows for tuning the balance between target search and coverage to reflect confidence in the learned semantic priority. The normalization of $p(f)$ addresses states where an important frontier only has a single non-zero feature or low feature activations in $\phi(f)$, which can lead to a low-valued semantic priority $p(f)$. By normalizing $p(f)$ by the maximum value in the current state, the combination of semantic priorities and coverage gains proposed in Equation (3.9), with a fixed $\alpha$ across different scenarios, becomes more robust.

### Plan Execution and Control

We now explain how the exploration planner navigates the robot through the environment, which is summarized in Algorithm 1. At every time step, the perception module updates the topological graph, the frontier set, and the robot's position. The tour is replanned whenever the current frontier set $\mathcal{F}_t$ or their coverage gains change (Algorithm 1). In that case, the priorities of all current frontiers are updated, and then a new tour $T$ is found by minimizing Equation (3.8) using a large neighborhood search (LNS) algorithm [77]. In each iteration, our custom LNS algorithm uses random destruction of up to 30% of the tour and reconstructs it using the cheapest insertion heuristic [78] follwed by a 2-opt swapping search [79]. Given a new tour, the next frontier in the tour is chosen as the subgoal $f_g = T(1)$. If the tour is not recomputed and the subgoal $f_g$ has been reached (Algorithm 1), the next node in $T$ is set as the goal. Otherwise, $f_g$ stays the same. The shortest path to $f_g$ is planned using A* [80] through $\mathcal{G}_t$, and the robot moves to the first node in the path $v_{p,1} \in \mathcal{V}_t$, applying $\mathbf{a} = \|v_{p,1} - v_t\|$.

Under the assumption of a perfect perception module that will correctly detect all frontiers within its range, our planning approach will eventually visit every frontier becoming available, independent of the priority function. Since only graph nodes with a minimum coverage gain are considered frontier viewpoints, tours will not include already visited frontiers, guaranteeing that the robot always moves towards unexplored spaces. Therefore, our planner can ensure complete exploration of the environment.

## 3.5 Experiments

### 3.5.1 Experimental Setup

Experiments are conducted in a Python-based 2D simulator with simplified sensing and navigation [62, 71]. Important aspects of the experiments are detailed below.

### Scenario Setup

We use ProcThor [81] to sample multi-room indoor floorplans and realistic object placements with 4 different room categories (kitchen, bathroom, living room, bedroom). We generate environments with 3 kitchens, 3 bathrooms, 1 living room, and 1 bedroom, arranged with constrained connectivity (bedroom only accessible from the living room, bathrooms to the living room via the kitchens). We configure two scenario setups with a different target object and starting room type, detailed in the following sections. Top-down maps and object data are extracted for the simulator, and additional small objects are sampled to increase semantic feature density. Scenarios are curated to ensure challenging tasks where semantic features offer an advantage for target search. Both setups use 30 scenarios for generating intervention datasets and 34 scenarios for obtaining the evaluation results.

### Oracle-based Data Generation

For training the semantic priority model, we generate synthetic interaction datasets by simulating expert interventions with an oracle model based on the expert model in Section 3.4.2. The oracle assigns priorities based on room types, favoring frontiers in target rooms or exploring unseen rooms. Rooms are classified using a list of characteristic object classes for each room category. We also generate a dataset with an exponential distance

Table 3.1: Model and oracle parameters.

| Parameter | | Oracle Value | Model Value |
|---|---|---|---|
| Irrational choice probability | $\rho$ | 0.0 | 0.1 |
| Rationality parameter | $\beta$ | 25.0 | 10.0 |
| Min. distance discounting factor | $\epsilon$ | 0.2 | 0.1 |
| Room feature vector weight | $\lambda$ | – | 0.7 |
| Exponential distance dicounting factor | $\gamma$ | 0.1 | – |
| Oracle intervention threshold | $\tau$ | 0.05 | – |

discounting model, that is $\delta(f) = \exp(-\gamma d'_t(f))$, as well as datasets with varied $\beta$ and $\tau$ to evaluate the robustness of our method to different expert behaviors (see Section 3.5.3). Finally, we vary the number of episodes $N_{eps}$ in the intervention dataset to evaluate the data efficiency of our method. The model parameters of the oracle are given in Table 3.1.

**Training**

The weights of the semantic priority model are trained using Adam [82] minimizing the negative log-likelihood of the observed expert choices (Equation (3.7)) for 2000 epochs with learning rate 0.01. For each dataset, the training uses 10 different random seeds.

### 3.5.2 Overview of Experiments

We evaluate the performance of our method in two task setups (see Section 3.5.1) and present both qualitative and quantitative results. A coverage baseline, similar to [18], uses the planner proposed in Section 3.4.3, but with the priority function $P(f) = I(f)$. For both task setups, we first present qualitative results to illustrate an example scenario and the behavior of our method and the baseline. Second, we evaluate the target search performance of our method using quantitative metrics and compare it to different oracle methods, serving as upper bounds for the search performance. Using the same metrics, we evaluate the robustness of our method to different expert datasets by varying the number of interventions and parameters of the oracle model in the first scenario setup.

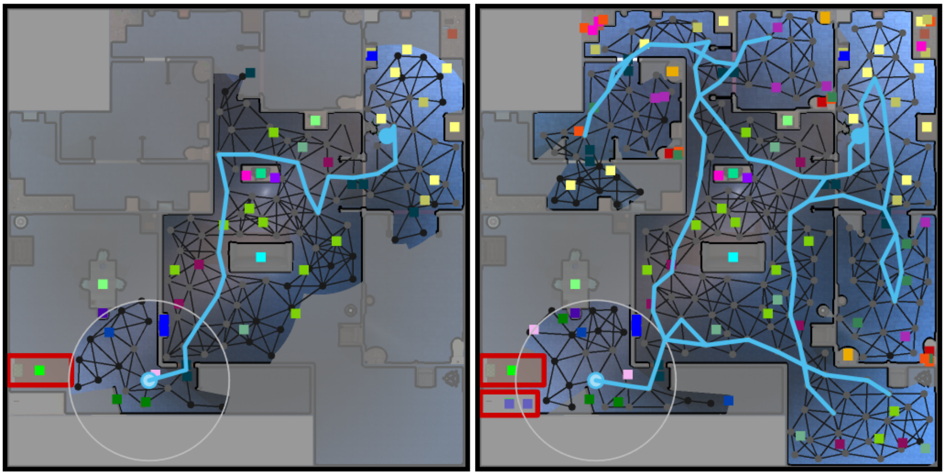
**Metrics**

We evaluate target search performance using the following metrics:

- *Path Length Ratio to Coverage (PLR)*: The episode-wise ratio of the path lengths $l$ until target discovery between the compared method and the coverage planner, i.e., PLR = $l_{\text{sem}}/l_{\text{cov}}$. The compared method reaches the target faster than coverage exploration for $PLR < 1$.

- *Success weighted by Path Length (SPL)*: The ratio of the traveled and shortest path to the target. A value of 1 indicates the shortest possible path to the target.

While the SPL metric is common in object search [83], the PLR metric is proposed as the main metric to evaluate the efficiency of our method compared to the coverage planner as it quantifies the relative advantage over coverage per scenario.



(a) Learned Priority Model
SPL = 0.784, PLR = 0.226

(b) Coverage Exploration
SPL = 0.177, PLR = 1.0

Figure 3.2: Top-down views of an example scenario of the first task setup comparing coverage-driven exploration with our learned semantic priority model. Frontier nodes of the topological graph are colored black, and others are gray. Blue edges visualize the path taken by the robot; the larger blue circle is the robot's position at target discovery time, and the smaller blue circle is the initial position. The red rectangles are target objects. Object instances are visualized as small squares colored according to semantic class.

### Oracle Methods

In our performance evaluation, we compare our method to the following oracle methods:

- *Oracle Interventions* waypoint interventions from the oracle model overwrite the coverage baseline behavior
- *Oracle Priorities* guides the planner with the semantic priorities from the oracle model.
- *Linear Oracle* uses a linear expert model (as Equation (3.2)) with hand-tuned weights to obtain semantic priorities.

## 3.5.3 Primary Scenario Results

In the primary scenario setup, the target object is a bed in the bedroom, and the robot is initialized in one of the kitchens. Therefore locating the living room first and then the door to the bedroom is necessary.

### Qualitative Results

Figure 3.2 compares the paths taken by the coverage planner and our target search planner with learned priorities in an example scenario (dataset $N_{eps} = 30$). The target object is in the bedroom (lower left) the robot starts in a kitchen (top right) and a large living room at the center connects the bedroom and kitchens. Figure 3.2 shows that our framework can guide the robot to the target object using a substantially shorter path than the coverage planner.

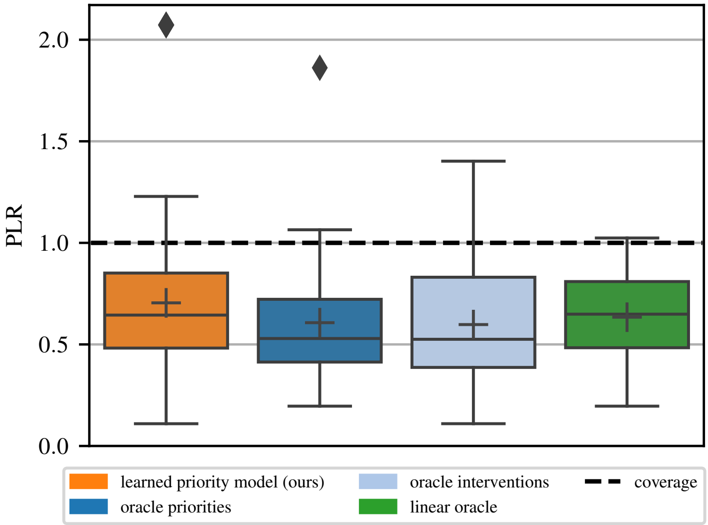


Figure 3.3: Performance results in the primary target search task: Comparison of our method to oracle methods, displaying the episode-wise path length ratio (PLR) to the coverage baseline (dashed line) as boxplots.

Initially, the robot navigates to the living room instead of exploring the other doorway below, as observed objects in the living room are prioritized. The robot discovers a higher density of relevant objects in the lower part of the living room in turns in that direction. The robot also prioritizes door objects to search for the bedroom, leading the robot to the correct target room. Finally, discovered bedroom objects yield the highest priority and lead the robot to the target object. Conversely, the coverage planner prioritizes frontiers only based on coverage gain and first explores the large open spaces in the living room and, subsequently, the smaller rooms, ignoring semantic features. These exemplary results illustrate that our framework can leverage semantic features in the environment to achieve better target search efficiency than coverage-driven exploration.

### Performance Results

We evaluate the target search performance of our method using $N_{eps} = 30$ in multiple test scenarios. The 340 episode results (10 training seeds and 34 test scenarios) are visualized as boxplot in Figure 3.3. The orange boxplot shows that our method significantly outperforms the coverage planner (dashed line) in most scenarios (median PLR = 0.644), up to a best-case performance of PLR = 0.11. In 88% of episodes, our method is more efficient than the coverage planner, and in 97% of the episodes, PLR is smaller than 1.3, indicating that cases where our method misguides the robot are rare. Moreover, our approach matches the linear oracle and is only slightly outperformed by the non-linear oracle guidance. These results show that our approach learned the underlying semantic priorities of the oracle expert and effectively leverages them in multiple unseen scenarios. That is, by incorporating the learned priorities in the cost function of the planner, it prioritizes exploration frontiers likely to lead to the target. Table 3.2 additionally reports the SPL metric indicating a strong advantage in absolute target search performance over coverage

Table 3.2: Comparison of our method with the coverage baseline and oracle methods across different metrics, defined in Section 3.5.2

| **Method** | **SPL (Task Setup 1)** | **SPL (Task Setup 2)** |
|---|---|---|
| Coverage Priorities | 0.406 ± 0.196 | 0.341 ± 0.313 |
| Oracle Priorities | 0.704 ± 0.202 | 0.564 ± 0.275 |
| Oracle Intervention | 0.712 ± 0.206 | 0.529 ± 0.281 |
| Linear Oracle Priorities | 0.650 ± 0.207 | 0.520 ± 0.271 |
| **Learned Priorities (ours)** | **0.627 ± 0.225** | **0.520 ± 0.313** |

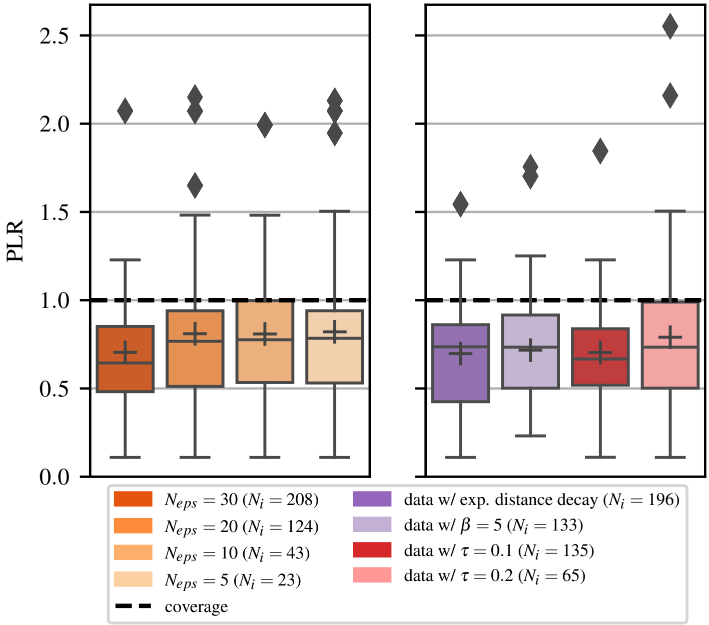
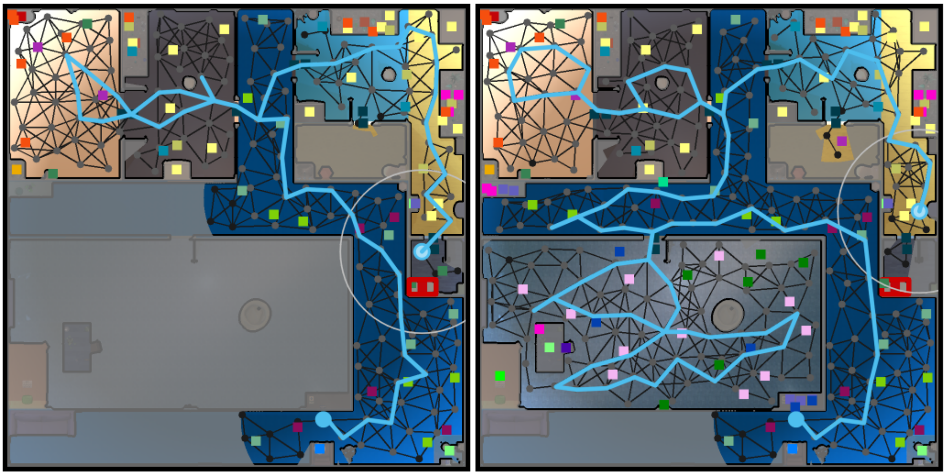


Figure 3.4: Comparison of different dataset sizes and oracle behaviors used for training, displaying PLR performance of the resulting priority models.

exploration and competitive performance compared to oracle methods.

### Robustness to Data Variation

Next, we analyze the robustness of our method to different dataset sizes $N_{eps}$ and expert behavior by varying the oracle parameters. For each dataset, semantic priority models are trained and tested as described in Section 3.5.3. Figure 3.4 shows the resulting PLR boxplots. The left subplot shows the results for a reduced number of training episodes, ($N_{eps} = 30$ is the same as in Figure 3.3). It is evident that with all 4 datasets, similar PLR performance is achieved. However, performance drops from $N_{eps} = 30$ to $N_{eps} = 20$, but further reduction up to $N_{eps} = 5$ does not affect the performance. Note that our method can achieve strong target search efficiency with only $N_i = 23$ expert interventions ($N_{eps} = 5$).



(a) Learned Priority Model
SPL = 0.509, PLR = 0.530

(b) Coverage Exploration
SPL = 0.270, PLR = 1.0

Figure 3.5: Top-down views of an example scenario of the second task setup comparing the behavior of coverage-driven exploration and our learned semantic priority model. Visuals follow the same conventions as in Figure 3.2.

A substantial improvement with more training data is only observed at $N_{eps} = 30$, which likely results from highly informative data points that only occur in this dataset, indicating that additional data can lead to further performance gains. The right subplot shows the results for 4 different oracle variations: exponential distance discounting instead of linear (Equation (3.3)), reduced expert rationality $\beta$ (increased noise, Equation (3.6)), and increased expert intervention threshold (less engaged, more selective expert), all with $N_{eps} = 30$. Our method is robust to these changes and yields similar results across all variations. The lowest performance occurs for $\tau = 0.2$, since a less engaged expert might miss providing some informative interventions.

## 3.5.4 Secondary Scenario Results

The secondary scenario setup uses the same maps as the primary, but the target object is a toilet in one of the three bathrooms, and the robot starts in the living room. Here, the robot must first prioritize finding any of the kitchens that will lead to the bathrooms and the target object. In this setup it is harder to leverage semantic features as two kitchen-bathroom pairs might attract the robot but do not yield the target.

### Qualitative Results

Figure 3.5 presents an example scenario of the secondary target search task, comparing the coverage planner with our planner guided by learned priorities. The target object is in the bathroom on the right side, and the robot starts in the bottom branch of the living room. The living room connects to a large bedroom in the center and 3 kitchen-bathroom pairs at the top of the map. The coverage robot incurs much performance loss when exploring the bedroom, while the learned semantic priorities favor continuing in the living room. Both remaining paths in the upper part of the map are very similar, as the

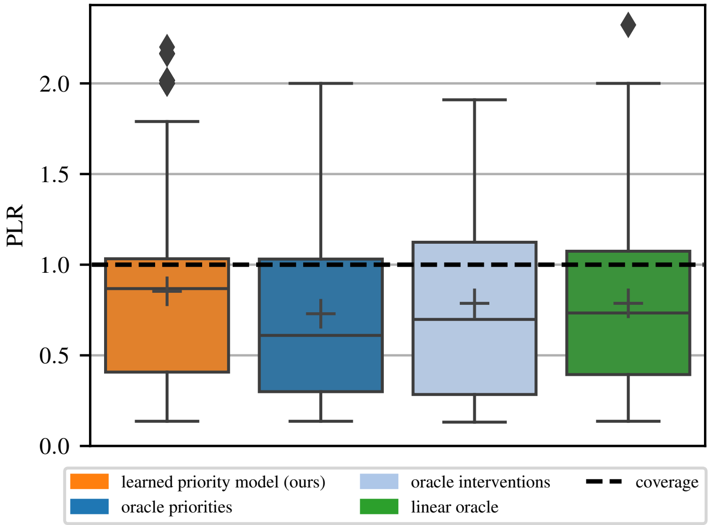


Figure 3.6: Performance results in the secondary target search task: Comparison of our method to oracle methods, displaying the episode-wise path length ratio (PLR) to the coverage baseline as boxplots.

semantic features cannot strongly favor one direction over the other; all small rooms are semantically promising. This example scenario indicates that the advantage of semantic over coverage exploration is less pronounced in this scenario setup, as only the bedroom is a clearly semantically irrelevant area, while the remaining rooms are all prioritized.

### Performance Results

Quantitative performance results in the secondary task setup are presented in Figure 3.6, analogous to Section 3.5.3. While our method outperforms the coverage planner (PLR < 1) in most episodes, the mean PLR of 0.853 is closer to 1 than in the primary task setup. This indicates more similar behavior of our method to the coverage planner, possibly as semantic priorities are less informative for target search. This is also supported by the PLR boxplots of the oracle methods, showing that more episodes perform similar to coverage than in the primary setup. Moreover, this task setup features a larger median gap between our approach and the oracle methods This shows that the difficulty of this task setup is exacerbated when using potentially noisy learned semantic priorities, giving more influence to the coverage gains in the tour cost function (Equation (3.9)). However, while some scenarios do not provide much room for improvement over coverage, the results show that our approach substantially improves target search efficiency in many other scenarios.

## 3.6 Conclusion

In this paper, we presented a novel approach to target search in unknown environments, combining semantic priorities learned from expert guidance with a global exploration planner. We trained the semantic priority model weighting exploration frontiers based on semantic features, such that a derived expert model matches a dataset of expert interventions. The combinatorial exploration planner prioritizes frontiers based on semantic priority and expected coverage gain, ensuring robust exploration independent of the learned model. The results show that the exploration planner guided by the learned priority model exhibits efficient target search behavior and outperforms a purely coverage-driven planner variant across different scenarios and simulated expert datasets. Future work will consider more realistic environments with complex semantic relationships and learning from real human data.

# 4

# Semantic Target Search and Exploration using MAVs in Cluttered Environments

*Autonomous target search is crucial for deploying Micro Aerial Vehicles (MAVs) in emergency response and rescue missions. Existing approaches either focus on 2D semantic navigation in structured environments, which is less effective in complex 3D settings, or on robotic exploration in cluttered spaces, often lacking the semantic reasoning needed for efficient target search. This chapter overcomes these limitations by proposing a novel framework that utilizes semantic reasoning to minimize target search and exploration time in unstructured 3D environments using an MAV. Specifically, the open vocabulary inference capabilities of Large Language Models are employed to embed semantic relationships in segmentation images. An active perception pipeline is then developed to guide exploration toward semantically relevant regions of 3D space by biasing frontiers and selecting informative viewpoints. Finally, a combinatorial optimization problem is solved using these viewpoints to create a plan that balances information gain with time costs, facilitating rapid location of the target. Evaluations in complex simulation environments show that the proposed method consistently outperforms baselines by quickly finding the target while maintaining reasonable exploration times. Real-world experiments with an MAV further demonstrate the method's ability to handle practical constraints like limited battery life, small sensor range, and semantic uncertainty.*

This chapter is based on: *N. Sethi*, **M. Lodel***, L. Ferranti, R. Babuska, J. Alonso-Mora, "STEM: Semantic Target Search and Exploration using MAVs in Cluttered Environments", under review. *Indicates equal contribution.*

*Statement of Contribution:* Max contributed the initial idea of extending existing frontier-based MAV exploration methods with semantic priorities and developed the target search planner. Under Max's supervision, Nikhil developed and implemented the pipelines for semantic priority masking and viewpoint evaluation. Max and Nikhil jointly contributed to the experiments and writing of the paper. Laura, Robert, and Javier provided discussions and feedback on the results and the manuscript.

## 4.1 Introduction

Micro Aerial Vehicles (MAV) are a promising tool to effectively search and explore complex unknown environments in domains such as search and rescue, inspection, and environmental monitoring. To relieve human operators from the challenging task of guiding MAVs through uncertain environments, methods for autonomous search of target objects are critical to improve the efficiency and effectiveness of such MAV missions.

In large and cluttered environments with many occlusions, the time efficiency of the search strategy is crucial due to limited available flight time and onboard sensor capabilities. Humans can efficiently search unknown spaces by leveraging their experience as well as semantic information, such as observed objects, to reason about the target's likely location. For example, dangerous chemicals are more likely to be found in a storage room than in an office. Building on this idea, recent work [16, 19, 20, 84–89] has shown that learning such semantic priors can significantly reduce the target search time by guiding the robot toward promising regions. However, as semantic observations or priors can be uncertain or unavailable in many real-world scenarios, purely relying on such semantic guidance may lead to inefficient behavior. Therefore, a robust target search strategy should balance semantic search and coverage-maximizing exploration to ensure an efficient and successful search.

Existing works on semantically-guided target search [16, 19, 20, 84–89] have focused on ground robots moving in 2D, while MAVs' 3D capabilities remain underexplored. In particular, MAVs can overcome occlusions in cluttered environments by changing their altitude and, therefore, improve search efficiency. Moreover, these target search methods rely on inferring semantic relationships from large pre-trained models [85], potentially generalizing poorly across highly specific scenarios, such as in disaster response. This underlines the need for integrating and balancing semantic search with efficient coverage-maximizing exploration approaches such as [9–11]. These methods achieve high coverage efficiency by leveraging long-horizon combinatorial planning techniques, but ignore semantic information that could guide the search towards the target.

In this work, we present STEM, a framework for Semantic Target Search and Exploration for MAVs. By building on recent advances in exploration planning and semantics-driven navigation, our framework enables both semantically guided and efficient exploration in complex 3D environments.

### 4.1.1 Related Work

In this section, we discuss existing approaches and how they relate to our work, starting with planning approaches for pure coverage exploration, and then focusing on target search approaches that leverage semantic information.

**Exploration Planning**

The problem of navigating a robot autonomously through an unknown environment to build a complete map from sensor observations has been investigated using a variety of different approaches. The fundamental idea of most approaches is to choose robot actions or plans such that future *viewpoints* efficiently minimize unknown space in the environment. Viewpoints are poses in 3D space from which the robot can observe the environment.

Sampling-based approaches randomly sample viewpoints in free space and evaluate their potential information gain about the map. Rapidly Exploring Random Trees (RRT) have been used to plan a local tree of informative viewpoints [90–93], with [93] integrating object search by prioritizing salient objects in the environment. However, the high computational costs limit the planning horizon, leading to greedy and inefficient exploration.

Frontier-based exploration methods focus on observing the boundary between known and unknown space, the *frontiers*, to incrementally reduce the unknown space. While early methods [17] just choose the closest frontier as the next observation target, recent work shows that selecting frontiers to maintain high flight speed improves exploration efficiency [94]. Similarly to sampling-based approaches, these methods lack long-horizon planning capabilities.

Recent works have shown that combining elements of both sampling-based and frontier-based exploration with planning can improve exploration efficiency. The approaches in [9–11, 95, 96] sample viewpoints around different frontiers, and then find a time-optimal global plan that connects these viewpoints using combinatorial planning. Following this approach, [9–11] plan over frontier viewpoints by solving a Travelling Salesman Problem (TSP). Specifically, the FUEL framework [9] considers drone dynamics in the TSP cost matrix to achieve efficient and agile exploration. In FAEL [95], a version of the Minimum Latency Problem (MLP) is used for planning that prioritizes frontier viewpoints with high coverage gains. Such approaches can be scaled to large environments using coarse global planning [11, 96].

The authors of FUEL [9] show that their approach achieves efficient and robust exploration performance on a real-world MAV platform in varying, complex 3D environments, due to an effective integration of global and local planning. As we are interested in 3D semantically-guided exploration with MAVs, we build on the FUEL framework [9] as exploration baseline, extending it with 3D semantic representations and planning to enable semantic target search. Our target search planner uses an MLP-based formulation similar to FAEL [95] and integrates semantic information to guide exploration toward target-relevant objects.

### Target Search

Autonomously searching for a target object in an unknown environment has been primarily investigated in the domain of indoor structured environments such as apartments, where clear semantic relationships between objects exist [16, 19, 20, 84–89].

These approaches differ in the source of learned semantic priors and the planning strategy used to guide the robot toward the target. In earlier works, domain-specific environment datasets are used for training navigation policies using reinforcement learning (RL) [16, 84] and training cost-to-go functions using self-supervised learning [19, 20]. Conversely, recent works [85–89] use foundation models such as Vision-Language Models (VLM) [97, 98] or Large Language Models (LLM) [21] trained on internet-scale data. The works [85–88] demonstrate zero-shot VLM/LLM-based target search in indoor environments using embedding similarity scores to choose exploration frontiers. In [85, 87, 88], the frontier selection is facilitated by propagating similarity scores into 2D [85, 87] or 3D [88] map representations. SEEK [89] proposes to distill semantic knowledge from an LLM into a lightweight model for efficient online inference.

The planning strategies used in most of these works are either based on learned navigation policies [16, 84] or greedy frontier selection [19, 20, 85, 86]. In contrast, [89] uses a Bayesian network prediction model and value iteration planning to choose the best region to search.

The main limitation of these methods is that they only consider greedy decision-making in 2D and structured indoor environments. We aim to fully make use of the MAV's 3D capabilities and address semantic uncertainty in unstructured environments. Our method builds on [86], owing to its simple approach to obtaining LLM-based quantitative semantic relationships, and extends it to a 3D planning pipeline that balances target search and exploration. That is, we propagate the semantic similarity scores into 3D space, similar to [88], and use a combinatorial planner that prioritizes viewpoints covering frontiers with high semantic similarity to the target.

### 4.1.2 Contribution

The main contribution of this paper is an active perception pipeline that can embed semantic priorities in 3D, generating a rich set of viewpoints with balanced coverage and semantic information gains. Using a combinatorial target search planner, efficient global plans through these 3D viewpoints are created that minimize the expected search time. Building on the FUEL framework [9], we provide a method for semantic frontier evaluation and a novel planner formulation that prioritizes viewpoints likely leading to the target based on semantic information. We conduct extensive experiments in both simulation and real-world environments using a Micro Aerial Vehicle (MAV) that validate the effectiveness of our approach. Our quantitative results show that our method consistently outperforms exploration-only baselines in terms of target search time and success rate.

## 4.2 Preliminaries

### 4.2.1 Problem Formulation

An MAV is tasked with exploring a previously unseen 3D environment represented as bounded volume $\mathcal{W} \subset \mathbb{R}^3$ to find a target object in minimum time. The MAV's pose in the environment at time instant $t$ is defined as $\mathbf{x}_t \in SE(3)$, and we assume that fast and accurate 3D position and attitude controllers are available, such that the robot can follow a trajectory by tracking pose increments [99]. It has a maximum linear velocity $v_{max}$, maximum acceleration $a_{max}$, and maximum yaw rate $\omega_{max}$. The robot is equipped with an RGB-D camera that provides a local observation of the environment. At each time instance $t$ the robot receives a measurement tuple $z_t = (\mathbf{x}_t, \mathcal{I}_c, \mathcal{I}_d)$, where $\mathcal{I}_c$ and $\mathcal{I}_d$ are the RGB and depth images, respectively.

Using the RGB images, the robot can perform semantic segmentation to identify objects in the environment. A set $\mathcal{S}$ of possible objects of interest (OOIs) represented by natural language semantic labels is available for segmentation. Importantly, it is assumed that the objects in $\mathcal{S}$ have semantic relationships defined by a function $F : \mathcal{S} \times \mathcal{S} \longrightarrow \mathbb{R}^+$, that quantifies how semantically related two objects in $\mathcal{S}$ are. It can be exploited to guide the robot towards a target object $o^* \in \mathcal{S}$. A target is considered found when its relative semantic segmentation area in the robot's field of view crosses a threshold $\lambda_{min}$.

**Problem Statement:** Given a target object $o^*$, a bounded volume $\mathcal{W}$, and the robot's initial configuration $\mathbf{x}_0$, find a collision-free global plan $\sigma$ through $\mathcal{W}$ at each time instant t such that $o^*$ is discovered in minimum time, using the history of observations $z_{0:t}$ and the semantic relationship function $F$.

## 4.2.2 Background

### Frontier Exploration Planning

The goal of robotic exploration is to efficiently build an occupancy map $\mathcal{M}$ of a bounded volume $\mathcal{W}$ using local range observations such as depth images. This map is a 3D volumetric grid of voxels, with each voxel $m_k \in \mathcal{M}$ storing the probability of occupancy $P_k$. These probabilities are updated using an inverse camera sensor model, and Bayesian Inference [100]. Building an occupancy map of an unknown environment requires the robot to reduce the unexplored space in $\mathcal{M}$.

Frontier-based exploration is an effective approach for reducing unknown space, which first detects a set $\mathcal{F}$ of *frontiers*, i.e., boundary voxels between known and unknown space. Then, it directs the robot to observe these frontiers efficiently. In this work, we build on the recent frontier-based MAV exploration method FUEL presented in [9], which is introduced briefly hereinafter.

To maintain efficiency, frontiers are clustered into groups of voxels using a region-growing algorithm, leading to a set of clusters $\mathcal{K}$, each with a minimum size $F_{min}$. That is, a cluster $K_i \in \mathcal{K}$ with $K_i \subseteq \mathcal{F}$ is only valid if $|K_i| \geq F_{min}$. Around these clusters, a set $\mathcal{V}$ of *viewpoints* is sampled, which are poses in free space that can 'view' the frontiers. Viewpoints are further filtered using a minimum information gain $I_{min}$, considering only those that view at least $I_{min}$ frontier voxels. Unknown space in $V$ can be reduced by finding the most efficient path through high-quality viewpoints. Recent frontier exploration methods such as FUEL [9] and FAEL [95] use combinatorial optimization methods to plan a non-myopic global path through the viewpoints in $\mathcal{V}$.

### Semantic Relationships

Semantics are labels or categories that humans use to classify objects. Humans use accumulated semantic knowledge to derive relationships between objects of interest when looking for targets. For instance, when searching for a `laptop`, we first look for a `table` as opposed to a `toilet` because the former is more correlated with the target object. These relationships are formalized by the semantic relationship function $F$ that maps a set $\mathcal{S}$ of semantic classes represented by language labels to scalar-valued similarity scores.

Large language models (LLM) or vision-language models like CLIP [101] or BERT [21] can use their contextual understanding to infer such relationships. These models use a neural network to first transform the text input to *vector embeddings*, which are real-valued representations of the text in a high-dimensional feature space. Objects that often occur close to each other often have similar vector embeddings, as they appear in similar contexts in the training data. Therefore, the cosine similarity score can be used to approximate the semantic relationship function $F$ between two labels, obtained by calculating the dot product of their vector embeddings **a** and **b**:

$$F(\mathbf{a},\mathbf{b}) = \frac{\mathbf{a}\cdot\mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|} \tag{4.1}$$

Such similary scores have been employed for target search in [85, 86].

# 4.3 Methodology

## 4.3.1 Overview

To motivate our method, consider how humans search for important objects. We infer target-object relations from the environment context, create a mental map of interesting objects, and then search *near* these OOIs to find the target. For instance, we might search near a wardrobe to find clothes or under a table to locate a person during an earthquake (see Fig. 4.1). Our method follows the same approach: To quickly find a target, it is essential to minimize unknown space *near* objects of interest that are conceptually and spatially related to the target object. Therefore, we *prioritize* search in specific regions, unlike coverage-based exploration, which tries to minimize *all* unknown space. Additionally, balancing both tasks is necessary to ensure that the method works even under semantic uncertainty and does not incur significant costs in exploration time. Our pipeline consists of three components.

- **Semantic Priority Masking**: This module processes the RGB image to segment objects, ranks them using an LLM, and compresses the segmentation image into a 2D priority mask (see Section 4.3.2 and Figure 4.1).
- **Active Perception**: This module uses the generated priority mask, the depth image, and the drone's state to (a) sample a set of 3D frontier viewpoints in free space, and (b) evaluate the semantic information gain of each viewpoint (Section 4.3.3).
- **Target Search Planner**: This module solves a combinatorial optimization problem over the 3D viewpoints to create a global plan that prioritizes high-gain viewpoints, minimizing the expected search time (Section 4.3.4).

Figure 4.2 visualizes the active perception and planning parts of the pipeline.

## 4.3.2 Semantic Priority Masking

The goal of this module is to use the RGB image $\mathcal{I}_c$ to generate a priority mask image $\mathcal{I}_p$ that has pixel-wise discrete priority values for each object of interest. A semantic segmentation image $\mathcal{I}_s$ and a set $\mathcal{S}_t \in \mathcal{S}$ of currently visible classes are generated using $\mathcal{I}_c$ at time $t$. The user-defined set $\mathcal{S}$ consists of a variety of relevant objects that can be encountered in the present scenario. In this work, we assume the existence of a learning-based method, such as Mask-RCNN [102] or Fast-SAM [103], that can generate $\mathcal{I}_s$ and $\mathcal{S}_t$.

Instead of using $\mathcal{I}_s$ directly, we compress the semantic segmentation image into a priority mask $\mathcal{I}_p$. This image contains pixel-wise discrete integers for each class, indicating their relative importance to the target class. The process to generate this mask is demonstrated in Fig. 4.1. A priority mapping function $r : \mathcal{S} \longrightarrow \mathbb{N}^+$ is derived *offline* as detailed below, and then queried online. The *priority mask* $\mathcal{I}_p$ is created at runtime by replacing each pixel in $\mathcal{I}_s$ with its corresponding priority value.

**Offline Priority Inference:** First, situational context about the scenario is added to the target object label in the set $\mathcal{S}$, using the formulation: `[label]` `[preposition]` `[context]`,

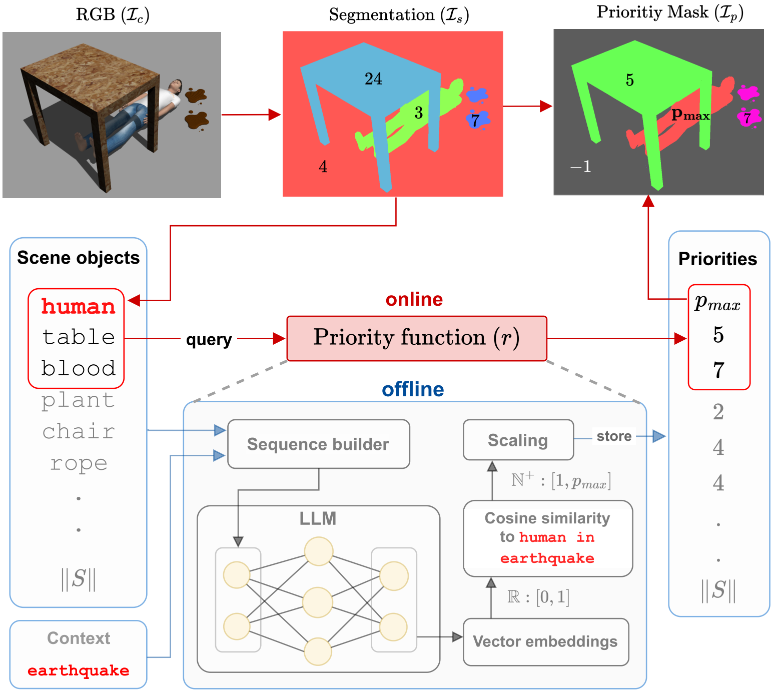


Figure 4.1: Semantic priority masking pipeline. Red arrows and blue arrows represent online and offline operations, respectively. At runtime, the priority of each class in $\mathcal{S}_t$ is queried from a pre-computed priority vector to create the priority mask $\mathcal{I}_p$.

e.g., `human in earthquake`. This helps to derive more accurate relationships between the target object and other environment objects. Second, each class in $\mathcal{S}$ is tokenized and passed through the LLM (bert-large-uncased [21]) to produce an output tensor of embeddings, which is averaged along the sequence dimension, resulting in $\tau$ of size $\|\mathcal{S}\| \times n_e$. Here, each row represents a class in the embedding space. The target embedding vector is $\tau^*$. To obtain cosine similarity scores for each class, each vector in $\tau$ is compared to $\tau^*$ using Equation (4.1), producing values between 0 and 1. Finally, the similarity scores are scaled to integer values within the range $[1, p_{\max}]$. Here the maximum priority value $p_{\max}$ is a tunable parameter controlling the sensitivity of the semantic search. The resulting priority mapping function $r$ from the set of classes to integer-valued priorities is stored offline, and then queried online with the set $\mathcal{S}_t$.

### 4.3.3 Active Perception

The goal of the active perception module is to use the priority mask $\mathcal{I}_p$, the depth image $\mathcal{I}_d$, and robot pose $\mathbf{x}_t$ to create a set of viewpoints $\mathcal{V}$ in free space and corresponding information gains $I$ for each viewpoint.

Section 4.3.3 provides a method for mapping of semantic priorities in 3D and Section 4.3.3 describes a method to diffuse semantic priorities to neighboring frontiers. Section 4.3.3 describes the process of generating viewpoints in free space and Section 4.3.3 de-

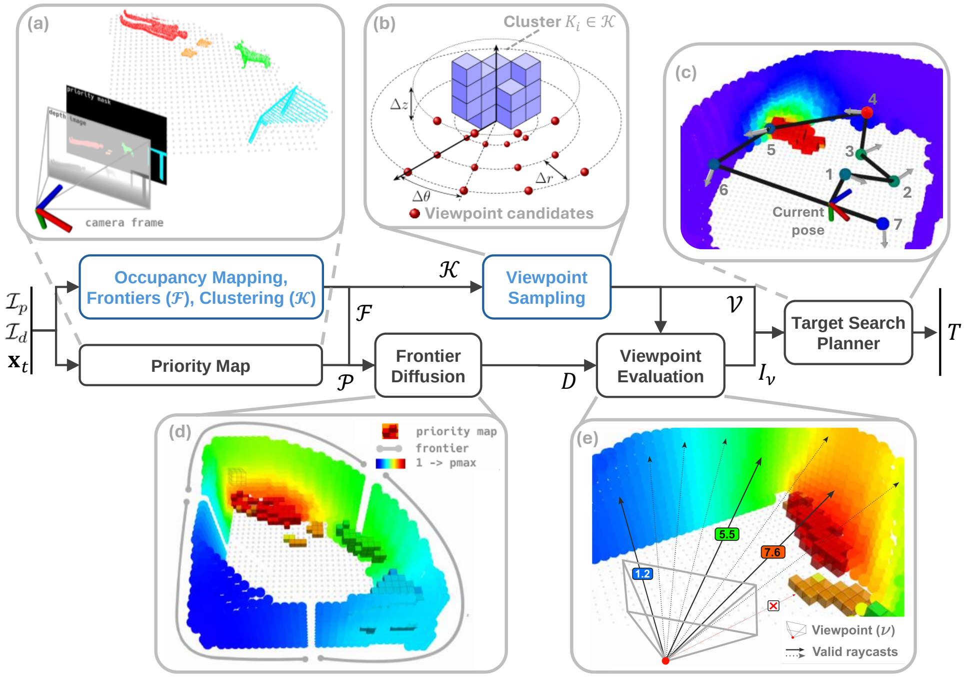


Figure 4.2: Active perception and planning pipeline. The blue blocks are from the FUEL framework [9], and Subfigure (b) is inspired by [9]. In (d), the priority map voxels are cubes, and frontier voxels are spheres.

scribes our procedure to calculate information gain for each sampled viewpoint. Figure 4.2 shows an overview of the complete active perception pipeline.

### Priority Map

The goal of this module is to represent the priority values in $\mathcal{I}_p$ in 3D space where the drone will subsequently navigate and collect new observations. The priority mask $\mathcal{I}_p$, the depth image $\mathcal{I}_d$, and robot pose $\mathbf{x}_t$ are used to create a 4D position-intensity point cloud observation $\Omega$. Each point $\Omega_k = [\mathbf{q}_w, p_w]^T$ in this point cloud is represented by the 3D position in the world frame $\mathbf{q}_w = [x_w, y_w, z_w]^T$ and the priority value $p_w$ as the intensity channel. Let $\mathbf{d_i} = [u_i, v_i]^T$ be a pixel in the depth image $\mathcal{I}_d$ with depth value $z_i = \mathcal{I}_d(\mathbf{d_i})$ and $p_w = \mathcal{I}_p(\mathbf{d_i})$ be the corresponding priority value. The point cloud $\Omega$ is generated using standard projective transformations using the camera's intrinsic and extrinsic parameters.

The point cloud is also post-processed using voxel-grid filtering and statistical outlier removal. The priority value $p_w$ at each 3D point $k$ in $\Omega$ is then used to update a discrete volumetric grid $\mathcal{P}$ at the corresponding voxel $p_k \in \mathcal{P}$ using a simple weighted update (Eq. 4.2)

$$p_k \longleftarrow (1-\alpha)p_k + \alpha p_w \quad \forall k \in \{1,...,|\Omega|\}. \tag{4.2}$$

Here $\alpha$ is a learning rate that updates the map progressively and prevents noise from being integrated, and $|\Omega|$ is the size of the point cloud. Finally, a local section $\mathcal{P}_l \subset \mathcal{P}$ of the priority map centered around the drone is retrieved to keep the computational load bounded.

### Frontier Diffusion

The motivation for this section directly draws from the goal of minimizing unknown space *near* objects of interest (Section 4.3.1). If frontiers can have increased importance near objects of interest, we can refine the search to interesting regions of the environment and find the target faster.

This idea is implemented by diffusing the priority values from the local map section $\mathcal{P}_l$ into neighboring frontier voxels using a 3D partial convolution. We use a partial convolution for the diffusion process as it normalizes over only the valid (non-empty) voxels in the sparse frontier structure [104]. A Gaussian kernel with spread $\sigma$, and size $W$ is used, thus making it a 3D Gaussian filter. Fig. 4.2 shows a simulation example from RViz, where the diffusion process is applied to 3D frontiers. The diffusion process is applied to each frontier voxel in a *local* region surrounding the drone to maintain computational efficiency.

The frontier diffusion module thus results in semantic priorities attached to each frontier voxel, that is, a frontier priority function $D : \mathcal{F} \longrightarrow [1, p_{\text{max}}]$. which can further be used for downstream tasks like informative path planning.

### Viewpoint Sampling

The goal of this section is to generate a set of viewpoints $\mathcal{V}$ which are candidate poses sampled in free space to 'view' the frontiers in $\mathcal{F}$. This module directly uses the approach from the FUEL framework [9], which we briefly describe here.

The viewpoint generation is based on the frontier clusters $\mathcal{K}$ introduced in Section 4.2.2. Since even small regions of space can hold significance in semantic target search, we set the minimum cluster size threshold used in FUEL to $F_{\text{min}} = 0$. Frontier viewpoints $v \in \mathcal{V}$ are generated by uniformly sampling free-space poses in a cylindrical coordinate system around the centroid of each frontier cluster (see Figure 4.2c). The yaw angle of each viewpoint is set to maximize the sensor coverage of the frontier cluster. For more details, see [9].

### Viewpoint Evaluation

This section describes a heuristic for computing a balanced coverage and semantic information gain for a viewpoint $v \in \mathcal{V}$, using the diffusion-based frontier priorities $D(f)$ of each frontier voxel $f \in \mathcal{F}$.

Consider Figure 4.2d, where the frontier voxels are colored based on their priorities. Rays are cast from a candidate viewpoint $v$ toward the voxels in $\mathcal{F}$ to determine the priority value at the ends of valid rays. A ray is considered valid when it is unobstructed by occupied or unknown space in $\mathcal{M}$. Voxels at the end of valid rays create a new *visible* frontier set $\mathcal{F}_v \subset \mathcal{F}$ for each viewpoint $v \in \mathcal{V}$. The frontier priority value $D(f)$ for each $f \in \mathcal{F}_v$ is then passed through a transfer function $\Phi$ and summed up to give the total information gain $I(v)$ of the viewpoint $v$, as in

$$\Phi(f) = \max\Big\{\exp\big(\gamma(D(f) - 1)\big), 1\Big\} \tag{4.3}$$

$$I(v) = \sum_{f \in \mathcal{F}_v} \Phi(f), \tag{4.4}$$

where $\gamma \in \mathbb{R}^+$ is a tuning parameter for the balance between semantic and coverage exploration.

To demonstrate this, consider high-priority frontier voxels near semantic objects (as in Figure 4.2d), which are exponentially weighted by the transfer function $\Phi$. Thus viewpoints oriented towards semantically meaningful regions of the 3D space achieve a high information gain $I_v$ and can be prioritized in semantic target search. Conversely, consider a frontier $f$ far from semantically interesting objects with the lowest priority value, i.e., $D(f) = 1$, resulting in a volumetric coverage gain $\Phi(f) = 1$. The parameter $\gamma$ controls the greediness towards semantic priorities: a higher $\gamma$ increases the difference between semantically relevant and irrelevant areas, while a lower $\gamma$ shifts the objective towards coverage exploration. For $\gamma = 0$, the viewpoint gain equals the number of covered voxels, i.e., $I(v) = |\mathcal{F}_v|$.

The described method for information gain evaluation can integrate semantic priorities with volumetric coverage and uses a tunable exponential weighting function to ensure that semantically relevant viewpoints are prioritized.

### 4.3.4 Target Search Planner

The goal of the global target search planner is to use the set of viewpoints $\mathcal{V}$ and their respective information gains $I$ generated in the active perception module together with the drone's state $\mathbf{x}_t$ to plan a global path that minimizes the time to find the target object.

Our approach extends the idea of combinatorial planning between different viewpoints [9, 95] to determine their optimal visitation order. While FUEL [9] uses a traveling salesman problem (TSP) that minimizes the total traveling distance, semantic target search needs to prioritize viewpoints with high semantic information gain, which are expected to be close to the target. To this end, we propose to formulate the combinatorial target search problem as a variant of the Minimum Latency Problem (MLP) [76] that minimizes the average waiting time, or *latency*, of multiple tasks, which in our case are the viewpoints. Specifically, our *weighted* MLP (WMLP) formulation prioritizes minimizing the latency of semantically promising viewpoints, using the information gains $I(v)$ as weights for each viewpoint $v \in \mathcal{V}$.

The WMLP is formulated over a modified set of viewpoints $\mathcal{V}^*$: Firstly, we only consider viewpoints with a minimum information gain $I_{\text{min}}$, i.e., $\mathcal{V}' = \{v \in \mathcal{V} \mid I(v) \geq I_{\text{min}}\}$, similar as done in [9]. Secondly, we add the drone's current pose $\mathbf{x}_t$, i.e., $\mathcal{V}^* = \mathcal{V}' \cup \{\mathbf{x}_t\}$, planning over $N = |\mathcal{V}^*|$ poses. The tour $T$ describes a visitation order of the viewpoints $v \in \mathcal{V}^*$, where $T(i) = v_j$ denotes the $i^{th}$ viewpoint in the tour being $v_j$, for $i, j \in \{0, \cdots, N-1\}$. By definition, $T(0) = \mathbf{x}_t$, as each tour starts at the drone's current pose. Let $\mathbf{C} \in \mathbb{R}^{N \times N}$ be a cost matrix quantifying the traveling time between viewpoints, where $C_{ij}$ is the time required to move from viewpoint $v_i$ to $v_j$. Then the WMLP is formulated as

$$\min_{T} \sum_{i=1}^{N} I(T(i)) \sum_{j=1}^{i} C_{T(j-1),T(j)}, \tag{4.5}$$

where the inner sum computes the latency until visiting the $i^{th}$ viewpoint $T(i)$, and the tour cost function is a weighted sum of these latencies using the viewpoints' information gains $I(T(i))$. This formulation extends the classical MLP with uniform weights to weighted task latencies.

Solving the problem in Equation (4.5) means that viewpoints $v$ with higher information gain $I(v)$ get scheduled earlier in the tour. Since the information gain was calculated by balancing coverage and semantic priorities in Section 4.3.3, this objective pursues both coverage and target search, and is thus more robust to semantic uncertainty.

While defining the cost matrix $\mathbf{C}$ using Euclidean distances is common in TSP formulations and robotic exploration [11, 95], it does not account for the complex dynamics of MAVs. Thus, we use the kinematic cost function from [9] to compute the traveling times in the cost matrix $\mathbf{C}$. $C_{ij}$ is then defined as the minimum time required to switch between two viewpoints $v_i, v_j \in \mathcal{V}^*$.

$$C_{ij} = \max\left(\frac{\text{length}(v_i^{\mathbf{p}}, v_j^{\mathbf{p}})}{v_{\max}}, \frac{|v_i^{\psi} - v_j^{\psi}|}{\omega_{\max}}\right) \tag{4.6}$$

Here, $v_i^{\mathbf{p}}$ is the 3D position, length$(v_i^{\mathbf{p}}, v_j^{\mathbf{p}})$ computes the path length between $v_i^{\mathbf{p}}$ and $v_j^{\mathbf{p}}$, and $v_i^{\psi}$ is the yaw angle of the $i^{th}$ viewpoint from set $\mathcal{V}^*$.

The objective in Equation (4.5) is approximately solved using a large neighborhood search (LNS) algorithm [77] that iteratively destroys and reconstructs the tour $T$. In each iteration, the custom LNS algorithm randomly removes up to 30% of the tour and then reconstructs it using the cheapest insertion heuristic [78] follwed by a 2-opt swapping search [79].

In summary, the tour $T$ minimizes the time to arrive at regions of the environment that are semantically important with respect to the target object, thus approximately minimizing target search time.

## 4.4 Experimental Setup

### 4.4.1 Simulation Environments

Two realistic simulation environments were used to evaluate the algorithm in the PX4-Gazebo SITL simulator[1]. The *Earthquake* is a custom environment (Fig. 4.3), and the *Cave* (Fig. 4.4) is a section of the 'Cave Circuit 02' world from the DARPA SubT Challenge[2].

A common superset of objects was used as semantic clues for both environments. This set combines object classes from the DARPA SubT challenge and common sense objects that are expected near a `human` target in search and rescue situations. We also included

[1] `docs.px4.io/main/en/simulation/ros_interface.html`

[2] `https://www.darpa.mil/program/darpa-subterranean-challenge`

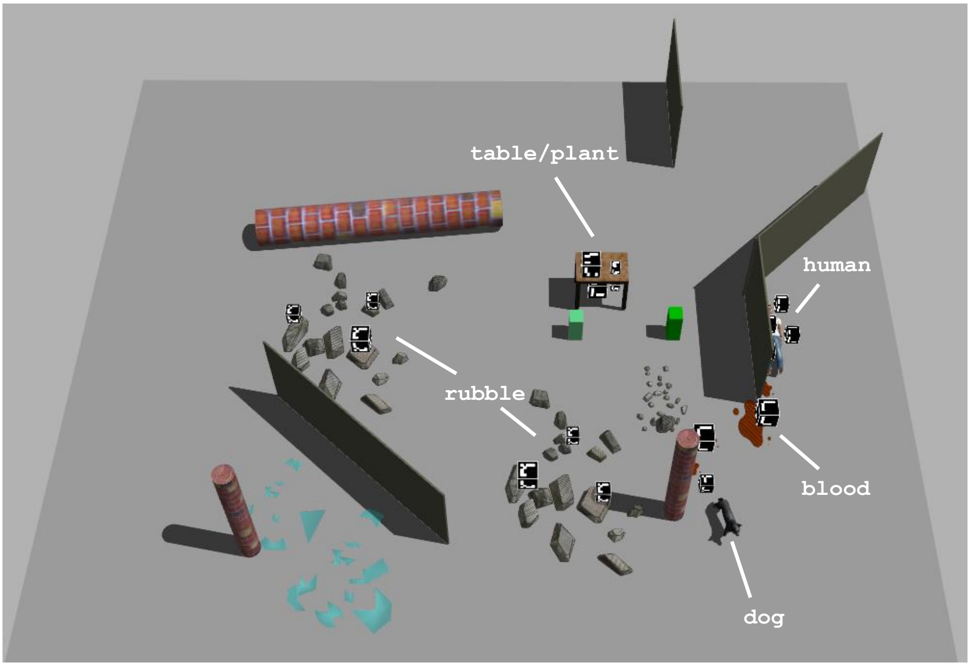


Figure 4.3: Gazebo environments (left: Earthquake, right: Cave).

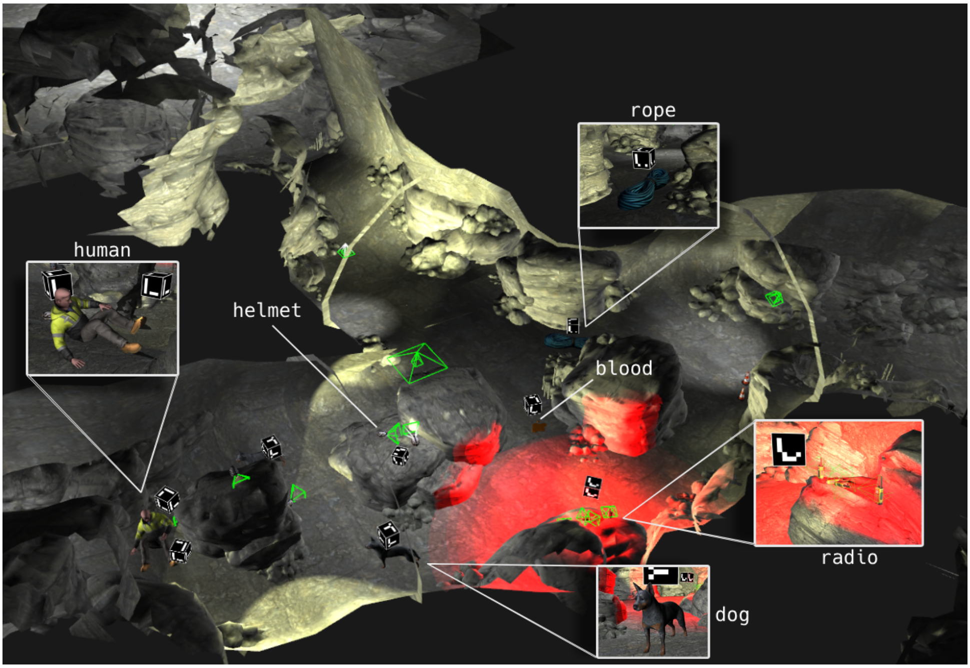


Figure 4.4: Cave environment (Gazebo).

Table 4.1: Parameter values for experiments. Note: (e.q. = earthquake).

| Parameter | | Value | Parameter | | Value |
|---|---|---|---|---|---|
| Priority update rate | $\alpha$ | 0.9 | Max. velocity | $v_{max}$ | 0.5 m/s |
| Max. acceleration | $a_{max}$ | 0.5 $m/s^2$ | Max. yaw rate | $\omega_{max}$ | 0.7 rad/s |
| Image height | $h$ | 480 | Image width | $w$ | 848 |
| Kernel spread | $\sigma$ | 2 | Kernel size | $W$ | 5 |
| Semantic greediness factor | $\gamma$ | 4 | Max. priority | $p_{max}$ | 8 |
| Min. viewpoint information gain | $I_{min}$ | 10 | Detection threshold | $\lambda_{min}$ | 0.01 |
| Number of object classes | $\|\mathcal{S}\|$ | 22 | Min. frontier size | $F_{min}$ | 0 |

unimportant distractor objects like `toy` and `plant` to evaluate whether the planner prioritizes semantically relevant objects rather than any observed object. Both environments contain a trapped human as the target, sufficiently occluded to make the problem challenging.

We use ArUco markers as a proxy for 2D image segmentation because existing detection pipelines performed poorly in a non-photo-realistic simulator like Gazebo. These markers were placed near their respective semantic objects, and the 2D segmentation image then contains a pixel-wise label for each marker $o \in \mathcal{S}$.

Both environments were made sufficiently large to ensure realistic exploration, and the start pose was chosen to be far from the target to observe the effect of semantic exploration. Since we evaluate the experiments on the absolute time-to-target metric rather than a relative metric (see Section 4.5.1), starting from random start poses in a relatively small environment will not give comparable results. However, it was observed that there is substantial variance in the SITL simulation (Section 4.4.2), primarily caused by non-deterministic communication between different software nodes, resulting in different observed trajectories for multiple runs with the same start pose. Therefore, we ran the same experiment multiple times for quantitative evaluation in each environment instead of varying the start pose.

### 4.4.2 Software Architecture

Our software stack is based on the Robotics Operating System (ROS) and integrates our target search method described in Section 4.3 with the modified FUEL exploration pipeline [9]. The architecture is demonstrated in Fig. 4.5. A key capability of the software is that we use the same pipeline for both hardware and software experiments, with the only difference being the source of the measurement tuple $z_t$. For simulation, this measurement comes from the Gazebo simulator, whereas for hardware experiments, this measurement comes from the onboard camera and motion-capture-based odometry. The parameters used by the planning pipeline are summarized in Table 4.1.

### 4.4.3 Hardware Setup

Hardware experiments were performed with a custom-built Micro Aerial Vehicle (MAV) previously used in [105] and modeled for the Gazebo simulation experiments. The MAV is equipped with an Intel Realsense D455 camera and an Nvidia Jetson Xavier NX on-

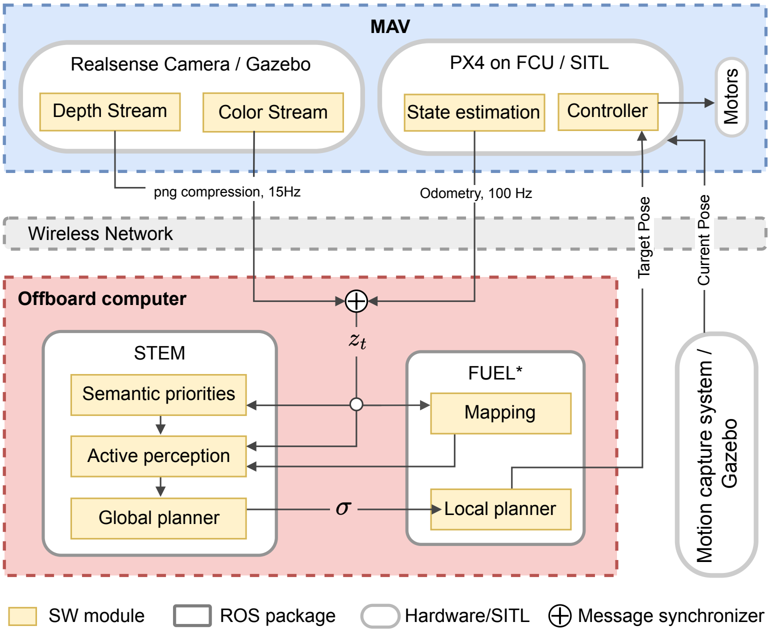


Figure 4.5: Software architecture. *A modified version of FUEL [9] is used for mapping and local planning.

board computer. A HolyBro Kakute F7 V2 flight controller was used with PX4 autopilot software. The MAV was localized in the environment via a Vicon motion capture system. ArUco markers were placed in the environment as semantic objects of interest, and the experiments were conducted in sufficiently cluttered configurations with screens and boxes as obstacles. Note that in the hardware experiments, we do not use LLM-based priority inference (see Section 4.3.3) but instead provide a handcrafted priority function $r$, to focus the experiments on the active perception and planning pipeline rather than the priority inference. Figure 4.6 shows a potential environment for the experiment.

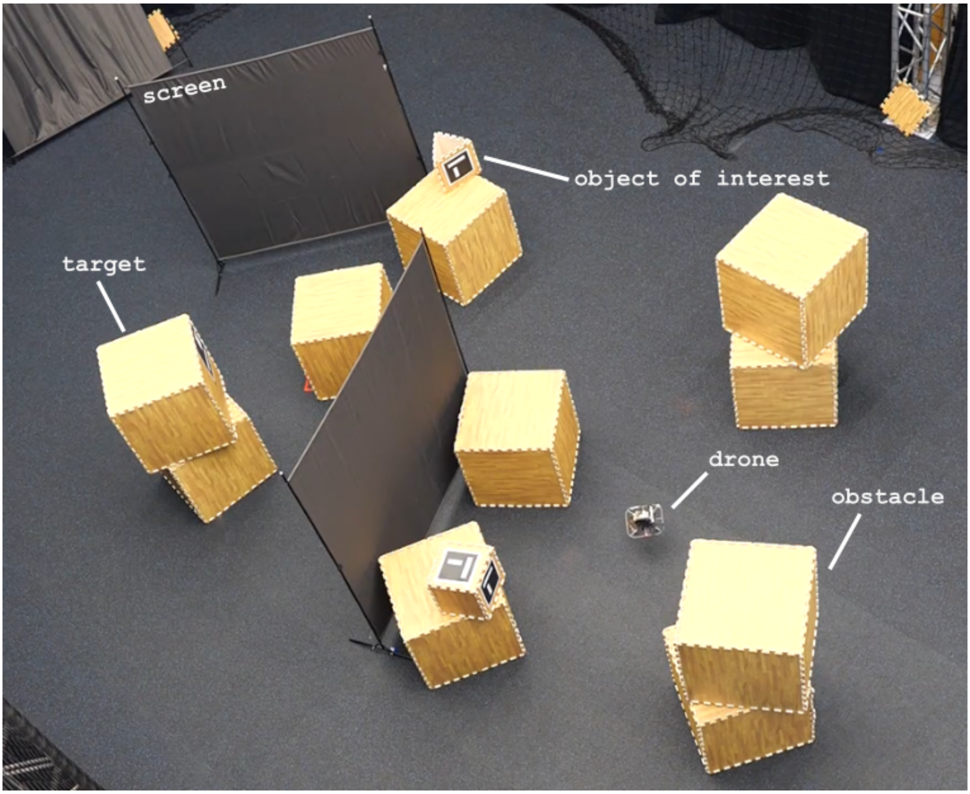


Figure 4.6: Lab environment for hardware experiments with example scenario.

## 4.5 Simulation Results

In this section, we present and discuss the results of our proposed method *STEM* on the simulation environments from Section 4.4.1. Evaluation metrics are proposed in Section 4.5.1, and two baselines are compared in Section 4.5.2. We provide the primary performance comparisons with baselines in Section 4.5.3, which are further supported using qualitative results in Section 4.5.4. Finally, we also conduct an ablation study to analyze the importance of the proposed target search planner in Section 4.5.5.

### 4.5.1 Evaluation Metrics

We employ the following metrics:

**Success %**: This metric calculates the percentage of episodes when the target was successfully found in a total of $n$ trials. An object $o$ is considered found when the fraction of object pixels $\lambda_o$ crosses a threshold $\lambda_{min}$ in the segmentation image, with $\lambda_o$ defined as $\lambda_o = \beta/(wh)$. Here, $\beta$ is the number of pixels belonging to the object's ArUco marker, and $w$ and $h$ are the segmentation image width and height, respectively. The parameter $\lambda_{min}$ depends on the environment complexity and camera intrinsics. For example, a value of $\lambda_o = 0.02$ means that the marker occupies 2% of the field of view in the image plane.

Table 4.2: Comparison study with baselines in the Earthquake and Cave environments. Two baselines from Section 4.5.2 were compared to our method on Success %, Time to target $t*$, and exploration time $t_f$.

| | | Target search | | Exploration |
|---|---|---|---|---|
| Env | Method | Success % | Time $t^*$ (s) | Time $t_f$ (s) |
| Earthquake | FUEL-original | 5% | 79.8 ± 0.0 | **107.3 ± 17.3** |
| | FUEL-complete | 100% | 76.6 ± 33.5 | 129.4 ± 12.9 |
| | **STEM (Ours)** | **100%** | **56.9** ± **22.8** | 139.9 ± 11.9 |
| Cave | FUEL-original | 40% | 65.0 ± 14.8 | **93.6 ± 30.7** |
| | FUEL-complete | 90% | 95.3 ± 26.6 | 166.4 ± 31.5 |
| | **STEM (Ours)** | **90%** | **64.1** ± **20.3** | 130.7 ± 19.8 |

**Time to target**: A commonly used metric for ObjectNav tasks is Success weighted by Path Length (SPL) [106]. A notable drawback of this metric is that it only considers travel distance in SE(3), and for robots with complex dynamics (like MAVs), the completion time is recommended [107]. Thus, we record the first time instant when a target $o^*$ was successfully detected (i.e., $\lambda_{o^*} \geq \lambda_{min}$) and call this metric as the Time to target $t^*$.

**Exploration time**: Since balancing exploration and target search is a secondary goal for our method, we also measure the exploration time $t_f$ in seconds. An environment is considered explored when no *visible* frontier can be found for 10 consecutive iterations. For a frontier to be considered *visible*, it must have (1) at least $F_{min}$ number of clustered voxels, and (2) at least one viewpoint with minimum information gain $I_{min}$. These conditions are also used by the authors of FUEL [9].

### 4.5.2 Baselines

Our method is compared to two versions of the FUEL exploration pipeline [9], differing by the two parameters $F_{min}$ and $I_{min}$: The *FUEL-original* baseline uses the original parametrization with $F_{min} = 100$ and $I_{min} = 20$ as proposed in [9]. For the *FUEL-complete* baseline, these parameters were tuned to maximize target search success, as we noticed that finding the target in small regions depended strongly on these parameters. For the Earthquake scenario, $F_{min} = 0$ and $I_{min} = 5$ were used, and for the Cave scenario, $F_{min} = 0$ and $I_{min} = 0$. Additionally, frontier down-sampling (see [9]) was turned off due to the narrow passages and small frontier sizes in the Cave environment. The sensor range $R_{max}$ for both baselines was kept the same as our method to make comparisons fair.

### 4.5.3 Performance Results

This section presents the performance results of our method compared to the two baselines introduced in Section 4.5.2 using the metrics from Section 4.5.1. Table 4.2 summarises results for both the Earthquake and Cave simulation environments. The data was gathered over 20 trials for each method and environment. The results are discussed hereinafter.

STEM consistently finds the target faster than all methods and is as successful as the FUEL-complete baseline, which is tuned for high success rates. Moreover, it keeps the exploration times within reasonable bounds. The fast target search times and high success rates have been achieved by the combination of the semantic viewpoint evaluation and the combined target search planner. The viewpoint evaluation based on semantic frontier priorities diffused from relevant objects and subsequent filtering of viewpoints using $I_{min}$ leads to a viewpoint set biased towards viewing regions of high likelihood of target presence. This allows our method to find the target consistently. The combinatorial target search planner schedules high semantic gain viewpoints earlier, such that the MAV quickly reaches regions where the target is likely to be found. Conversely, FUEL-complete does not use semantic information to guide search, and explores irrelevant regions first, therefore taking more time to find the target while still achieving high success rates.

The performance difference between FUEL-complete and STEM underlines the importance of the viewpoint gain threshold $I_{min}$: A lower $I_{min}$ retains viewpoints for small frontier clusters in tight spaces, and therefore enabling a high success rate for FUEL-complete. However, this also means that many small but unimportant regions are covered, leading to inefficient and slow target search and exploration, as reflected by the exploration time results of FUEL-complete. This emphasizes the advantage of evaluating the semantic relevance of frontiers, as it allows retaining and prioritizing viewpoints in small but important regions while ignoring small, unimportant regions.

FUEL-original rarely finds the target due to its increased minimum frontier size $F_{min}$ and viewpoint threshold $I_{min}$. This causes frontier clusters or viewpoints in tight spaces that lead to the target being ignored. However, this baseline consistently completes exploration faster, since the clustering and thresholding facilitate more stable viewpoint sets $\mathcal{V}$ and more consistent global plans, allowing the MAV to maintain high speeds throughout the episode. Furthermore, FUEL-original solves a metric TSP using the LKH heuristic solver, generating robust and efficient global plans.

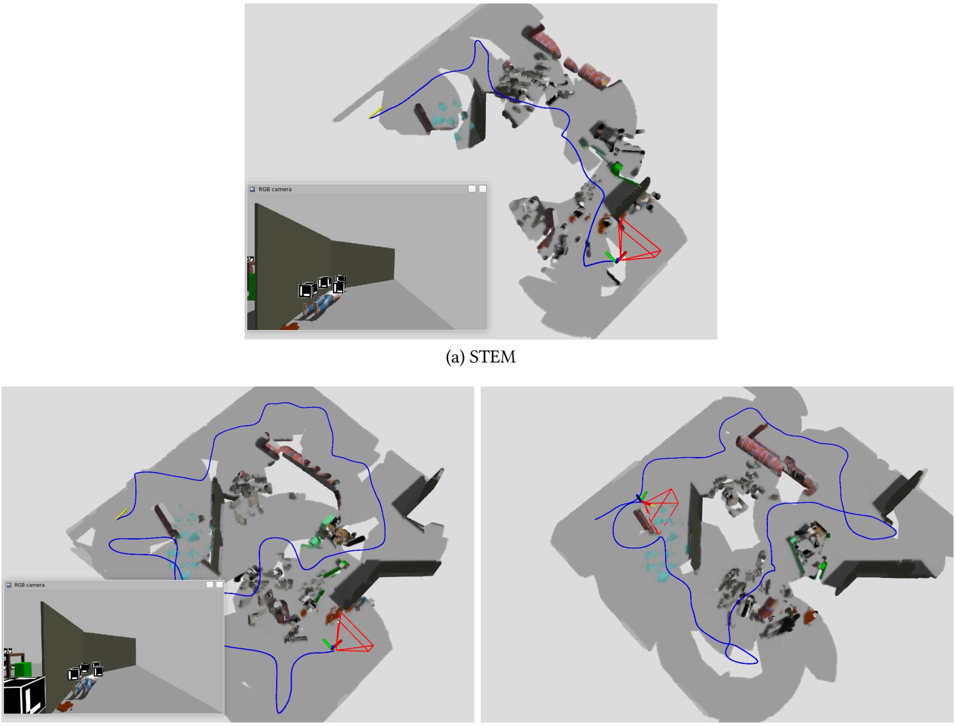
### 4.5.4 Qualitative Results

In this section, the behavior of our proposed method compared to the baselines is visualized to support the quantitative results from Section 4.5.3.

Figure 4.7 shows a qualitative comparison of the trajectories produced by the three methods in the Earthquake environment until the target is found or exploration is completed. The results show that STEM finds the target quickly without taking large detours, while FUEL-complete explores large parts of the environment before finding the target, and FUEL-original does not find the target. STEM achieves these results by focussing exploration on regions semantically related to the target, while both FUEL methods only aim at maximizing coverage gains. The trajectory of FUEL-complete shows, that it came close to the target and highly relevant semantic cues (`blood` objects) but does not react to these cues and continues exploring irrelevant regions.

Comparing FUEL-complete and FUEL-original emphasizes the effect of the thresholds $F_{min}$ and $I_{min}$ discussed in Section 4.5.3. When FUEL-original does not find any sufficiently large frontiers or viewpoints and completes exploration, there are still several small regions of the map left unexplored. FUEL-complete, however, is more thorough in its exploration but explores the environment less efficiently, as evidenced by more directional changes in its trajectory.

Figure 4.8 further shows keyframes of our algorithm performing target search in the earthquake environment. The MAV starts at a disadvantaged position and starts to explore



(a) STEM

(b) FUEL-complete

(c) FUEL-original

Figure 4.7: Qualitiative comparison with baselines for the Earthquake environment. Episodes were recorded until $t^*$ or $t_f$, whichever comes first. The reconstructed point cloud from RViz is shown along with the drone's trajectory (in blue). The camera FOV is shown in red, and the latest RGB camera image is displayed.

first to gather information. When it comes across semantically relevant objects such as `rubble` or `blood`, it samples informative viewpoints covering the nearby frontiers. Planning a path through such viewpoints continually using the target search planner allows the MAV to find the hidden target quickly.

Figure 4.9 presents the trajectories of our method and the two baselines in the cave environment. The results show that STEM is able to handle a complex 3D environment with many occlusions and tight spaces, which underlines its ability to balance exploration and semantic target search. After exploring initially, the MAV comes across objects such as `radio` and `dog` in the right arm of the cave, which STEM uses to guide the robot to the target quickly. Conversely, FUEL-complete misses the semantic cues and first explores the left arm of the cave before eventually discovering the target. FUEL-original coincidentally explores the right arm of the cave first but does not find the target due to its thresholding of frontier clusters and viewpoint gains.

### 4.5.5 Planner Ablation Study

In this section, we evaluate the importance of the combinatorial target search planner in our method. We compare our full method with the WMLP-based planner from Sec-

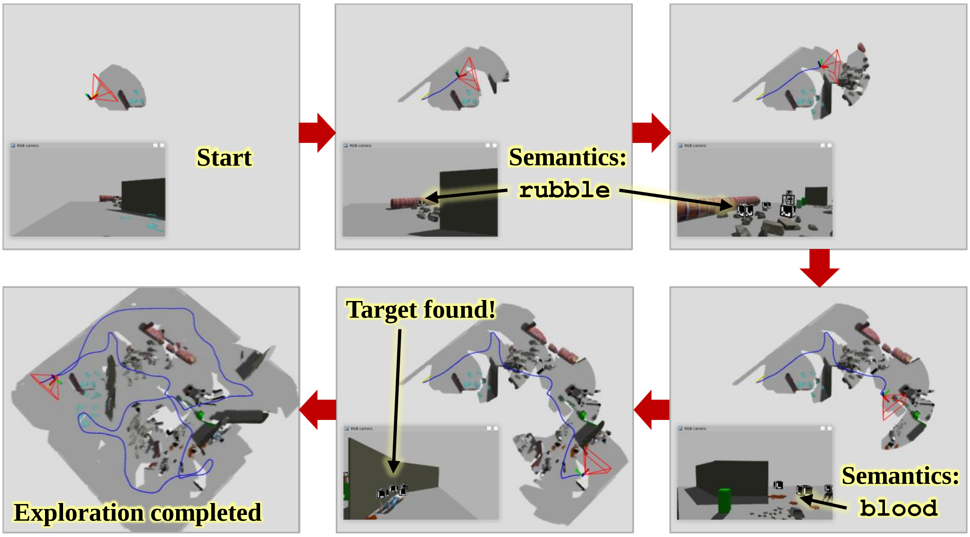


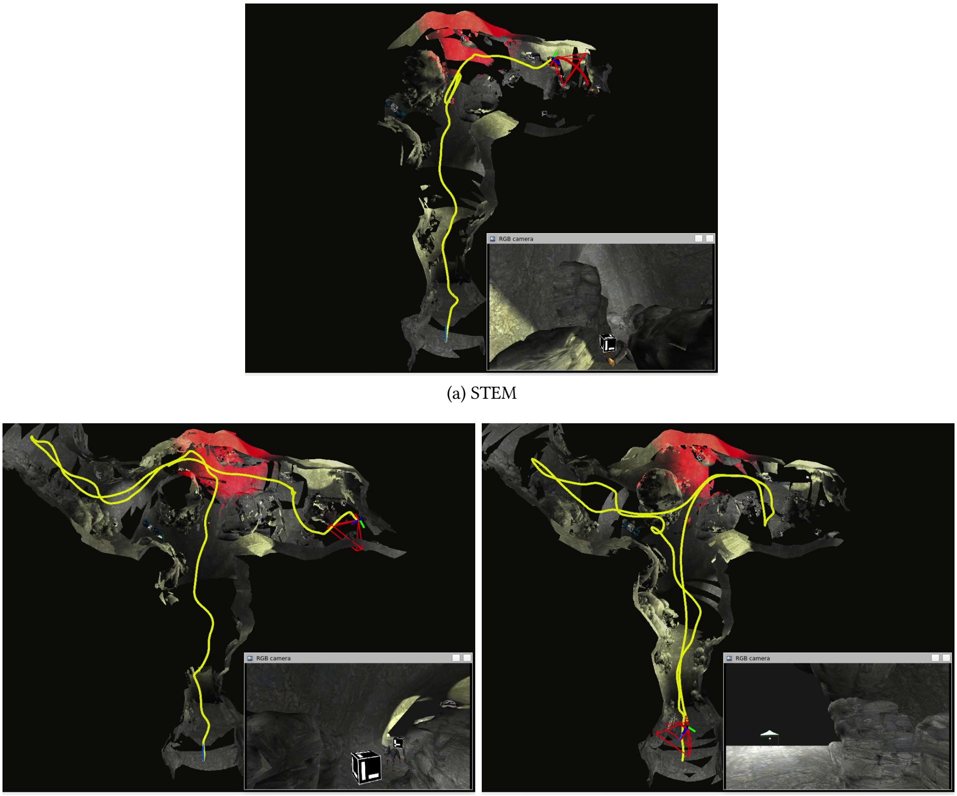
Figure 4.8: Key frames of target search using STEM in the Earthquake environment. The snapshots show the recorded RGB point cloud, the camera FOV in red, the trajectory in blue, and the current RGB image.

Table 4.3: Quantitative results of the planner ablation study, with two simplified variants of our WMLP-based planner: a greedy planner and a TSP-based planner. The metrics are the same as in Table 4.2.

| | | Target search | | Exploration |
|---|---|---|---|---|
| Env | Method | Success % | Time, $t^*$(s) | Time, $t_f$(s) |
| Earthquake | Greedy | 100% | $71.1 \pm 39.1$ | $210.5 \pm 25.8$ |
| | TSP-LKH | 100% | $65.3 \pm 26.7$ | $128.5 \pm 10.8$ |
| | WMLP (STEM) | 100% | $56.9 \pm 22.8$ | $139.9 \pm 11.9$ |
| Cave | Greedy | 50% | $64.9 \pm 18.1$ | $175.8 \pm 35.8$ |
| | TSP-LKH | 85% | $93.7 \pm 19.9$ | $134.1 \pm 23.5$ |
| | WMLP (STEM) | 90% | $64.1 \pm 20.3$ | $130.7 \pm 19.8$ |

tion 4.3.4 with two simplified variants that use the same viewpoint sampling, evaluation, and filtering methods using semantics but replace the planner. The first variant uses a greedy viewpoint choice, planning to the viewpoint with the highest information gain $I_v$. The second variant uses the TSP planner from [9] to plan a path through the viewpoints in $\mathcal{V}$, deploying the LKH heuristic to solve the optimization problem. The quantitative results are shown in Table 4.3.

The results show that STEM with the WMLP planner outperforms both the TSP and greedy planners on target search metrics in both environments. In the earthquake environment, all three variants achieve a 100% success rate, while in the cave environment, the success rate of the greedy planner is substantially lower than that of WMLP and TSP. This indicates that the complex geometry of the cave environment requires more consistent and



(a) STEM

(b) FUEL-complete

(c) FUEL-original

Figure 4.9: Qualitiative comparison with baselines for the cave environment. Episodes were recorded until $t^*$ or $t_f$, whichever comes first. The reconstructed point cloud from RViz is shown along with the drone's trajectory (in yellow). The camera FOV is shown in red, and the latest RGB camera image is displayed.

efficient behavior of the combinatorial planners than the simpler earthquake environment.

The average target search times $t^*$ show significant differences between the planners in both environments. In the earthquake environment, the TSP planner is 15% slower, and the greedy planner is 25% slower in finding the target than the WMLP planner. This underlines that the WMLP planner prioritizing semantically relevant viewpoints contributes substantially to STEM's target search performance. However, it also shows that the performance gap between FUEL and STEM (Section 4.5.3) is only partially caused by the planning algorithm, and that the semantic diffusion-based viewpoint evaluation and filtering are also critical elements of the target search strategy. In the cave environment, the greedy planner achieves equally fast search times as WMLP *when successful*, while the TSP planner is 46% slower. This indicates that in some cases, the semantic viewpoint gains used by the greedy planner help to find the target quickly (when relevant cues are observed), while the TSP planner is only optimizing for efficiently covering the given viewpoints.

While WMLP and TSP show very similar exploration times, the greedy planner is significantly less efficient in covering the volume. This is due to the greedy planner's tendency to oscillate between viewpoints with high information gain, leading to long detours and inefficient exploration paths.

In summary, this ablation study shows that combinatorial planning contributes to more consistent performance for both target search and exploration and that both the semantic viewpoint evaluation and the WMLP planner contribute substantially to the target search performance of STEM.

## 4.6 Real-world experiments

This section presents the results of real-world experiments conducted using the software architecture from Section 4.4.2 and the hardware setup from Section 4.4.3.

The results are visualized in Figures 4.10 to 4.12, showing our algorithm performing simultaneous target search and exploration in complex 3D environments set up in the laboratory. Figure 4.10 shows a scenario where the drone starts from the bottom right corner in the top-down view, and explores the environment searching for the target and relevant objects, represented by ArUco markers. The target is hidden behind a screen and a lower wall in the upper left section of the lab. The onboard camera images in Figure 4.10 show how the drone first discovers a high priority object on top of the wall, and therefore prioritizes the unknown area behind the wall due to the diffusion method (see Section 4.3.3). In image 2 a high relevance object is detected behin the wall, creating additional high priority frontiers around the pillar in the upper left corner, which leads the WMLP planner to prioritize exploring the area behind the wall and screen further. Therefore the drone is able to discover the target quickly and without large detours in image 3. Following target discovery, the framework guides the drone to further explore the environment efficiently, facilitated by the viewpoint evaluation method combining coverage and semantic information gain (see Section 4.3.3). This scenario underlines how our framework is able to search for a target using semantic clues in a cluttered 3D environment, guiding the drone to fly over obstacles (the wall close to viewpoint 2) to explore important hidden areas behind it.

Figures 4.11 and 4.12 show two additional scenarios, where the drone is able to find the target quickly using the AruCo marker objects as semantic clues, as described above. Both experiments show how our framework is able to guide the drone through different cluttered environments, handling occlusions and narrow passages, and to efficiently explore the environment after finding the target.

In summary the real-world experiments proved that our method is able to handle noisy onboard depth and RGB camera images, and still efficiently guide the drone towards semantically important areas. This enables the drone to find the target quickly in three different complex 3D environments, and to explore the environment efficiently after target discovery.

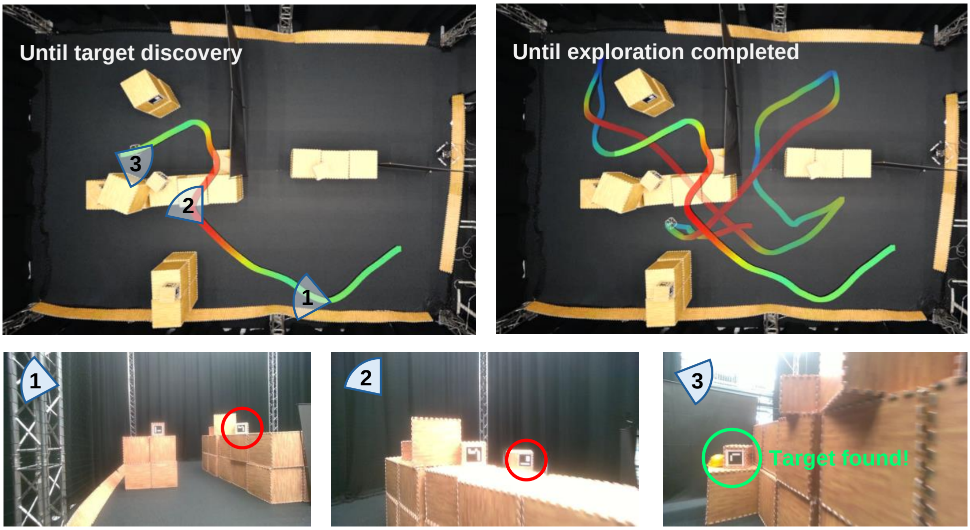


Figure 4.10: Hardware experiment 1 in the laboratory. Above two top-down views are shown, on the left with the trajectory until target discovery, on the right with the full trajectory until exploration is completed. The color of the trajectory indicates the height $z$ of the drone. Below images from the onboard RGB camera from three time instances are shown, which are marked by position and orientation in the left top-down view. The red circles in the onboard images indicate the high priority objects detected by the framework, that guide the drone towards the target. The last onboard image shows the moment the target is discovered.

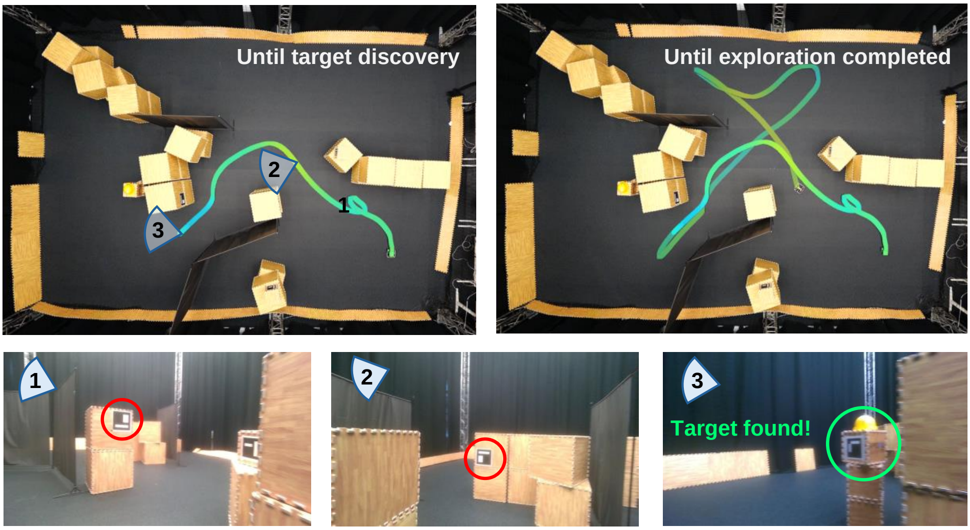


Figure 4.11: Hardware experiment 2 in the laboratory, with the same format as Fig. 4.10.

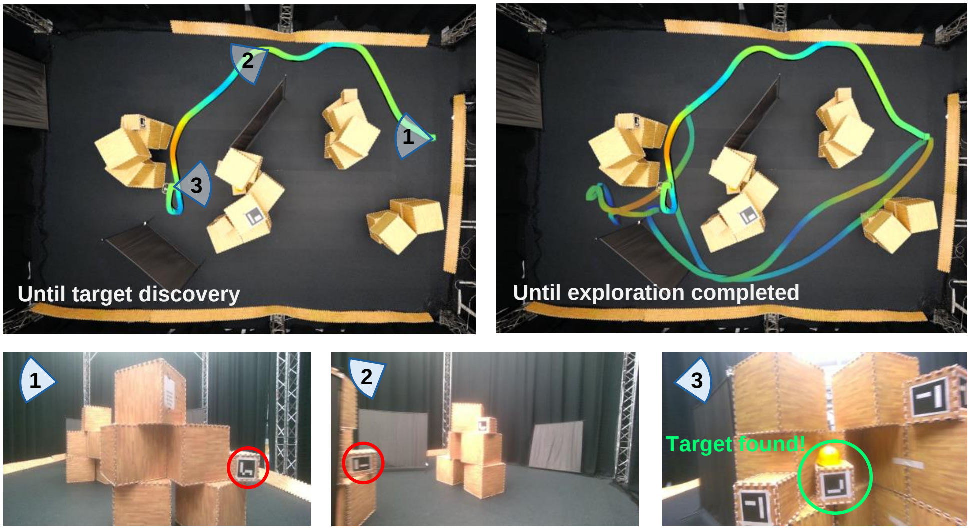


Figure 4.12: Hardware experiment 3 in the laboratory, with the same format as Fig. 4.10.

## 4.7 Conclusion

In this paper, we introduced STEM, a novel framework for semantically-guided target search and exploration with Micro Aerial Vehicles (MAVs) in complex, cluttered 3D environments. The core contribution of this work is the integration of learned semantic reasoning for target search with active perception and planning techniques commonly used in coverage exploration. The key idea of our approach is to prioritize exploration around semantically relevant objects when planning over all available exploration frontiers. Balancing semantic and coverage objectives in this way, efficient behavior for both target search and exploration is achieved.

Quantitative evaluation in two challenging simulation environments demonstrates that STEM achieves consistently higher success rates of more than 90 % and up to 32% faster target discovery times compared to coverage-maximizing baseline methods. Real-world experiments further validate the practical applicability of our approach, proving that our approach can exhibit effective target search with noisy camera inputs and realistic drone dynamics.

Future work will investigate how uncertainty about semantic object detection and semantic relationships can be propagated into the active perception pipeline, and how foundation models can be used to infer semantic priorities beyond embedding similarities.

# 5

# Conclusions and Future Work

*The final chapter of this thesis summarizes and concludes on the developed methods, the presented results, and the key findings. Subsequently, we discuss the limitations of our work and the open challenges and questions left for future research, focusing on multi-robot target search, semantic uncertainty, and modeling human expert knowledge.*

## 5.1 Conclusions

This thesis presents planning algorithms that enable mobile robot navigation in search and reconnaissance missions. Specifically, the problem of autonomously searching for a target in an unknown environment is addressed at the local and global planning levels. The first goal of the thesis was to develop an informative local motion planner that can navigate the robot safely through cluttered environments while minimizing the uncertainty about target locations using sensor observations. Local motion planning is limited by its planning horizon, which can lead to inefficient behavior in large, complex environments. Therefore, the second goal was to develop a global path planner that can guide the robot to search the entire environment efficiently. Such a planner must also account for the human operator's understanding of the scene as represented by semantic features and direct the search toward relevant regions. Consequently, the third goal was to develop a method for learning a model that predicts the semantic priority of different regions from user feedback to improve the efficiency of the global planner. Finally, the fourth goal was to demonstrate the global target search planner on a UAV platform, extending the planner to handle 3D cluttered environments and limited sensing capabilities.

### 5.1.1 Informative Local Motion Planning

Chapter 2 presented an informative trajectory planning framework that produces safe local motion plans such that sensor observations maximize the information about the target locations. The main challenge addressed in this work is the computational complexity of sampling future sensor observations for many candidate trajectories, which is infeasible for real-time planning. Prior approaches either resort to short-horizon, myopic planning or non-myopic but coarse path planning without kinodynamic feasibility. Therefore, we proposed a hierarchical framework where a viewpoint policy guides the robot towards informative observations, and an MPC-based motion planner ensures safe and feasible trajectories. The policy is trained using DRL to produce local 2D reference viewpoints that maximize cumulative mutual information about potential target locations. The MPC planner then generates a trajectory towards the reference viewpoint that adheres to kinodynamic and collision avoidance constraints. Crucially, the viewpoint policy is trained with the MPC planner as part of the environment dynamics, enabling the policy to specifically learn how to effectively guide the MPC planner. We trained and tested the proposed framework in simulation and evaluated its performance by time until completion in previously unseen environments. The results in the most challenging scenarios showed that our viewpoint policy outperformed a greedy viewpoint selection baseline with 9.4 % faster completion. This indicated the advantage of the policy's learned non-myopic reasoning. However, our approach was outperformed by a non-myopic MCTS baseline (13.3 % faster completion) that has privileged access to the global map. Despite this advantage, our approach achieved a lower number of failed episodes than both baselines. Importantly, inference of the viewpoint policy can be run at a rate three orders of magnitude faster than the MCTS planner, allowing for quick reactions to new observations and real-time operation.

### 5.1.2 Learning Semantic Target Search from Expert Guidance

Chapter 3 presented a framework for global target search planning prioritizing semantic features preferred by an expert. Previous semantics-driven target search approaches depend on large domain-specific datasets typically unavailable for tasks like search and rescue, while efficient coverage exploration planners neglect semantic information. We proposed to guide a combinatorial exploration planner with a semantic priority function that predicts how likely exploring a region will lead to the target. The semantic priority function is learned from recorded exploration waypoints provided by an expert, modeling the expert choices based on semantic features. The semantic priority predictions then guide the exploration planner, formulated as a weighted minimum latency problem. The planner aims to minimize the time to visit frontiers with both high semantic priority and coverage gain. We evaluated the proposed framework in randomly generated 2D multi-room simulation environments with different objects in each room and generated synthetic datasets of expert inputs using an oracle model. We compared the performance of the proposed framework and of the oracle model with a coverage exploration planner that does not consider semantic features. The results showed that our method reached the target faster in around 8 out of 10 episodes, and in the best case, achieved a traveling distance nine times shorter than the coverage baseline. Moreover, the performance has been competitive with the oracle model, which has access to the true semantic priorities. Our method has also been robust to variations in the dataset of expert inputs, specifically for small amounts of input data, increased noise, and different expert behavior.

### 5.1.3 3D Semantic Target Search with MAVs

Chapter 4 presented a framework for semantic target search and exploration using MAVs in cluttered environments. This chapter bridges the gap between prior work on semantics-driven target search on the one hand and the results achieved by 3D exploration planners for MAVs on the other. Existing target search methods lack the ability for viewpoint planning in 3D environments, and existing 3D exploration planners do not consider semantic features. Moreover, this chapter proposed an alternative to learning a priority model from user inputs (as in Chapter 3) and instead uses a pre-trained LLM to predict semantic priorities using embedding similarities. Our framework propagates the semantic priorities of observed objects into nearby frontier voxels and aggregates these frontier priorities by evaluating the information gains of sampled frontier viewpoints, combining semantic priority and coverage gain. The viewpoint gains are used in a combinatorial exploration planner similar to Chapter 3, prioritizing viewpoints covering many high-priority frontier voxels. Therefore, the time to find the target is minimized while maintaining efficient exploration. Efficient and smooth planning for MAVs is achieved by considering the kinodynamic constraints of the robot in the planner cost matrix. We evaluated the proposed framework in Gazebo simulations and real-world experiments. Simulations were performed in two unstructured 3D environments, where multiple semantic cues are available to guide the MAV towards the target. We evaluated our framework quantitatively by comparing the target search time and success rate to two coverage-only exploration baselines tuned for thorough and rapid exploration behavior, respectively. Compared to the thorough exploration baseline, we achieved an equal success rate but 25-32% faster target discovery, while compared to the rapid exploration baseline, we observed a 2.5-20 times higher success rate. At the same time, our framework does not sacrifice exploration efficiency, achieving similar or better full exploration times than the thorough exploration baseline. We conducted real-world experiments in three cluttered scenarios in a lab environment, where the MAV is tasked to find a hidden target object using available semantic features. The recorded behavior of our framework emphasizes its ability to direct the search towards semantically promising regions while using the MAV's 3D motion capabilities to efficiently search and explore the environment.

## 5.2 Future Work

This thesis contributed to navigation methods for searching and exploring unknown environments with mobile robots, combining planning and learning approaches. Nevertheless, there are a number of open challenges and opportunities for future research in this area that could improve the feasibility and effectiveness of deploying autonomous mobile robots in sensitive search missions. This section presents several potential research directions that can extend the work presented in this thesis.

### 5.2.1 Multi-Robot Target Search

When exploring large environments, speed and battery range can become limiting factors for deploying a single robot in a time-critical mission. Dividing the exploration task among multiple robots is, therefore, a common approach for efficient coverage exploration [108–110], but leveraging semantic features in multi-robot search is an open challenge. Given that the approaches presented in Chapter 3 and Chapter 4 separate the problems of semantic reasoning and efficient planning, a simple, centralized extension to multiple robots can be achieved by only modifying the planning part. As the current approach uses a combinatorial approach to planning tours across frontiers, the problem can be extended to a variant of the vehicle routing problem (VRP), where each robot's tour covers a subset of the frontiers, prioritizing frontiers with high semantic importance.

A shortcoming of this naive multi-robot extension is that the exploration workload is not shared efficiently between robots since only assigning viewpoints to robots does not consider the amount of area to be explored behind each frontier. Recent works [108, 109] effectively address this issue by partitioning the full unknown space, assigning the resulting subregions to robots in a higher-level planner, and then performing single-robot viewpoint planning inside partitions. To extend this approach to semantic target search, a key challenge is how to propagate and aggregate semantic information over subregions to prioritize those most relevant to the task. Moreover, in a decentralized multi-robot setting, sharing semantic information between robots is a challenge, as semantic voxel maps are memory-intensive. Both challenges can be addressed by developing suitable semantic environment representations. Semantic scene graphs, such as Hydra [111], enable a semantic representation of the environment at different levels of abstraction, such that object-level semantic information can be aggregated to a room or general subregion level. Then, only the relevant abstraction level needs to be shared between robots, reducing the memory requirements for communication.

Future work should investigate how to devise multi-level scene representations that can be effectively used for multi-robot target search and exploration planning, and how semantic priorities of subregions can be inferred from aggregated semantic information.

### 5.2.2 Semantic Uncertainty

All results presented in this thesis make significant simplifying assumptions about the environment and the robot's sensors and capabilities. Specifically, Chapters 2 and 3 show only simulation results with perfect object detection, and Chapter 4 shows experiments with a real drone but in a controlled lab environment using AruCo markers as objects. An open problem for real-world deployment is to explicitly model and handle uncertainty in the semantic features and relationships used in reasoning and planning for target search.

Both methods using semantic features in Chapters 3 and 4 rely on methods for semantic scene understanding, such as using RGB images for object detection and semantic segmentation. However, these methods are sensitive to noise in the sensor data, such as lighting conditions, occlusions, and motion blur [112–114]. As semantic features can be critical for effective target search, uncertainty in these features should be handled in both the created environment representations and derived planning algorithms for robust target search. On the side of representations, this problem is extensively investigated in the field of semantic mapping and semantic SLAM [111, 115], where semantic voxel maps or scene graphs can store information about the uncertainty of semantic labels. On the side of planning, Chapter 2 and various prior works [36, 116, 117] shows how uncertainty can be explicitly minimized in the planning problem by optimizing for an information-theoretic objective.

Future research should investigate how to integrate such an active classification objective into the target search planning problem, for example, by sampling additional viewpoints close to relevant and uncertain objects and including an additional information gain term for classification in the viewpoint evaluation. Moreover, such active scene understanding can extend to labels of the subregions (e.g., rooms) mentioned in the previous section.

### 5.2.3 Foundation Models and Expert Knowledge

In Chapters 3 and 4, we have proposed two different approaches to obtain a semantic priority model for guiding the target search planner. In Chapter 3, the semantic priority model is learned from expert inputs, while in Chapter 4, the semantic priority is derived from similarities between word embeddings produced by an LLM. The expert user model is only trained for a specific target search task to keep the number of required user inputs low. The LLM-based predictions depend on the availability of a large amount of text data relevant to the target search task during training. The major advantage of foundation models, such as LLMs, is that they have common-sense knowledge [118] from being trained on a wide range of human-made text and preference data. Common-sense knowledge is relevant even for highly specific tasks (e.g., knowing that blood stains might lead to an injured victim in a search and rescue scenario). However, it is not sufficient to make informed decisions in complex scenarios, as they lack domain-specific knowledge and operator preferences. Therefore, a critical challenge for future work to address is how the common-sense knowledge of foundation models can be leveraged while adapting them to specific tasks using expert inputs. A fundamental question here is which user input modalities are most effective both for the user to communicate their intentions and preferences and for the model to adapt to these inputs.

A possible approach to extend the concept of expert inputs by waypoint guidance in

Chapter 3 toward leveraging foundation models is to use an LLM to generate commonsense guidance inputs and pre-train a light-weight model on these inputs, similar to [89]. This model can then be fine-tuned for specific tasks using expert inputs and used for online inference to guide the target search planner.

Another approach is to use natural language instructions to specify task-specific knowledge, i.e., the user describes the scenario and his objectives and reasoning. When querying the model with observations in the environment, both the instructions and the query are used to predict where the target is most likely to be found. While [119] uses such an approach to greedily guide the robot from the language output, the approach employed in Chapter 4 uses embedding similarities to predict the semantic priority of different regions. Both approaches are limited in effectively capturing the model's reasoning to guide a planner.

Future work should investigate if an approach similar to [118], where a custom output layer for the LLM is trained to predict a semantic priority score given the language input, can be adapted for guiding a target search planner such as proposed in this thesis. Open questions for such an approach also include how to enable online inference with the typically computationally expensive LLMs and how to feed information about the environment from structured representations, such as scene graphs [120], into the LLM to reason about the environment and the target search task.

# Bibliography

[1] Antonio Loquercio, Elia Kaufmann, René Ranftl, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. Learning high-speed flight in the wild. *Science Robotics*, 6(59):eabg5810, 2021.

[2] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47):eabc5986, 2020.

[3] 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.

[4] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10965–10975, June 2022.

[5] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16901–16911, June 2024.

[6] Devendra Singh Chaplot, Dhiraj Gandhi, Abhinav Gupta, and Ruslan Salakhutdinov. Object Goal Navigation using Goal-Oriented Semantic Exploration. *Advances in Neural Information Processing Systems*, 2020-Decem, 2020.

[7] Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. Vlfm: Vision-language frontier maps for zero-shot semantic navigation. In *International Conference on Robotics and Automation (ICRA)*, 2024.

[8] Muhammad Fadhil Ginting, Sung-Kyun Kim, David D. Fan, Matteo Palieri, Mykel J. Kochenderfer, and Ali-akbar Agha-Mohammadi. SEEK: Semantic Reasoning for Object Goal Navigation in Real World Inspection Tasks, 2024.

[9] Boyu Zhou, Yichen Zhang, Xinyi Chen, and Shaojie Shen. Fuel: Fast uav exploration using incremental frontier structure and hierarchical planning. *IEEE Robotics and Automation Letters*, 6(2):779–786, 2021.

[10] Zehui Meng, Hailong Qin, Ziyue Chen, Xudong Chen, Hao Sun, Feng Lin, and Marcelo H. Ang. A Two-Stage Optimized Next-View Planning Framework for 3-D Unknown Environment Exploration, and Structural Reconstruction. *IEEE Robotics and Automation Letters*, 2(3):1680–1687, July 2017. Conference Name: IEEE Robotics and Automation Letters.

[11] C. Cao, H. Zhu, Z. Ren, H. Choset, and J. Zhang. Representation granularity enables time-efficient autonomous exploration in large, complex worlds. *Science Robotics*, 8(80), 2023.

[12] Bruno Brito, Boaz Floor, Laura Ferranti, and Javier Alonso-Mora. Model predictive contouring control for collision avoidance in unstructured dynamic Environments. *IEEE Robot. Autom. Lett.*, 4(4):4459–4466, 2019.

[13] Bruno Brito, Michael Everett, Jonathan P. How, and Javier Alonso-Mora. Where to go next: learning a subgoal recommendation policy for navigation in dynamic environments. *IEEE Robot. Autom. Lett.*, 6(3):4616–4623, 2021.

[14] Sihao Sun, Angel Romero, Philipp Foehn, Elia Kaufmann, and Davide Scaramuzza. A comparative study of nonlinear mpc and differential-flatness-based control for quadrotor agile flight. *IEEE Transactions on Robotics*, 38(6):3357–3373, 2022.

[15] Tao Chen, Saurabh Gupta, and Abhinav Gupta. Learning Exploration Policies for Navigation. *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[16] Nuri Kim, Obin Kwon, Hwiyeon Yoo, Yunho Choi, Jeongho Park, and Songhwai Oh. Topological Semantic Graph Memory for Image-Goal Navigation. In *6th Annual Conference on Robot Learning*, October 2022.

[17] Brian Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*, pages 146–151. IEEE, 1997.

[18] Junlong Huang, Boyu Zhou, Zhengping Fan, Yilin Zhu, Yingrui Jie, Longwei Li, and Hui Cheng. FAEL: Fast Autonomous Exploration for Large-scale Environments With a Mobile Robot. *IEEE Robotics and Automation Letters*, 8(3), 2023.

[19] Santhosh Kumar Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. PONI: Potential Functions for ObjectGoal Navigation with Interaction-free Learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18868–18878, 2022.

[20] Meera Hahn, Devendra Singh Chaplot, Shubham Tulsiani, Mustafa Mukadam, James M Rehg, and Abhinav Gupta. No RL, No Simulation: Learning to Navigate without Navigating. In *Advances in Neural Information Processing Systems*, volume 34, pages 26661–26673. Curran Associates, Inc., 2021.

[21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[22] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[23] 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 *Robotics: Science and Systems XIX*, 2023.

[24] Nils Wilde, Dana Kulic, and Stephen L. Smith. Active Preference Learning using Maximum Regret. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[25] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep Reinforcement Learning from Human Preferences. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[26] Yulun Tian, Katherine Liu, Kyel Ok, Loc Tran, Danette Allen, Nicholas Roy, and Jonathan P. How. Search and rescue under the forest canopy using multiple UAVs. *Int. J. Rob. Res.*, 39(10-11):1201–1221, 2020.

[27] Farzad Niroui, Kaicheng Zhang, Zendai Kashino, and Goldie Nejat. Deep Reinforcement Learning Robot for Search and Rescue Applications: Exploration in Unknown Cluttered Environments. *IEEE Robotics and Automation Letters*, 4(2), 2019.

[28] Marija Popovic, Gregory Hitz, Juan Nieto, Inkyu Sa, Roland Siegwart, and Enric Galceran. Online informative path planning for active classification using UAVs. In *2017 IEEE Int. Conf. Robot. Autom.*, pages 5753–5758, 2017.

[29] Antonio Vasilijević, Dula Nad, Filip Mandi, Nikola Miškovi, and Zoran Vukić. Coordinated navigation of surface and underwater marine robotic vehicles for ocean sampling and environmental monitoring. *IEEE/ASME Trans. Mechatronics*, 22(3):1174–1184, 2017.

[30] Mikko Lauri and Risto Ritala. Planning for robotic exploration based on forward simulation. *Robotics and Autonomous Systems*, 83, 2016.

[31] Graeme Best, Oliver M. Cliff, Timothy Patten, Ramgopal R. Mettu, and Robert Fitch. Dec-mcts: Decentralized planning for multi-robot active perception. *Int. J. Rob. Res.*, 38(2-3):316–337, 2019.

[32] Timothy Patten, Wolfram Martens, and Robert Fitch. Monte Carlo planning for active object classification. *Auton. Robots*, 42(2):391–421, 2018.

[33] Nikolay Atanasov, Jerome Le Ny, Kostas Daniilidis, and George J. Pappas. Information acquisition with sensing robots: Algorithms and error bounds. In *2014 IEEE Int. Conf. Robot. Autom.*, pages 6447–6454, 2014.

[34] Brent Schlotfeldt, DInesh Thakur, Nikolay Atanasov, Vijay Kumar, and George J. Pappas. Anytime planning for decentralized multirobot active information gathering. *IEEE Robot. Autom. Lett.*, 3(2):1025–1032, apr 2018.

[35] Chao Cao, Hongbiao Zhu, Howie Choset, and Ji Zhang. TARE: A Hierarchical Framework for Efficiently Exploring Complex 3D Environments. In *Robotics: Science and Systems XVII*. Robotics: Science and Systems Foundation, 2021.

[36] Benjamin Charrow, Gregory Kahn, Sachin Patil, Sikang Liu, Ken Goldberg, Pieter Abbeel, Nathan Michael, and Vijay Kumar. Information-theoretic planning with trajectory optimization for dense 3D mapping. In *Robot. Sci. Syst. XI*, volume 11, 2015.

[37] Ajith Anil Meera, Marija Popovic, Alexander Millane, and Roland Siegwart. Obstacle-aware adaptive informative path planning for UAV-based target search. In *2019 Int. Conf. Robot. Autom.*, pages 718–724, 2019.

[38] Heejin Jeong, Brent Schlotfeldt, Hamed Hassani, Manfred Morari, Daniel D. Lee, and George J. Pappas. Learning Q-network for active information acquisition. In *2019 IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pages 6822–6827, 2019.

[39] Delong Zhu, Tingguang Li, Danny Ho, Chaoqun Wang, and Max Q.H. Meng. Deep reinforcement learning supervised autonomous exploration in office environments. In *2018 IEEE Int. Conf. Robot. Autom.*, pages 7548–7555. IEEE, 2018.

[40] Alberto Viseras and Ricardo Garcia. DeepIG: Multi-robot information gathering with deep reinforcement learning. *IEEE Robot. Autom. Lett.*, 4(3):3059–3066, jul 2019.

[41] Kyle D. Julian and Mykel J. Kochenderfer. Distributed wildfire surveillance with autonomous aircraft using deep reinforcement learning. *J. Guid. Control. Dyn.*, 42(8):1768–1778, aug 2019.

[42] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to Explore using Active Neural SLAM. In *8th International Conference on Learning Representations, ICLR 2020*, 2020.

[43] Hai Zhu and Javier Alonso-Mora. Chance-constrained collision avoidance for MAVs in dynamic environments. *IEEE Robot. Autom. Lett.*, 2019.

[44] Amarjeet Singh, Andreas Krause, Carlos Guestrin, and William J. Kaiser. Efficient informative sensing using multiple robots. *J. Artif. Intell. Res.*, 34:707–755, 2009.

[45] Frédéric Bourgault, Alexei A. Makarenko, Stefan B. Williams, Ben Grocholsky, and Hugh F. Durrant-Whyte. Information based adaptive robotic exploration. In *IEEE Int. Conf. Intell. Robot. Syst.*, volume 1, pages 540–545, 2002.

[46] Héctor H. González-Baños and Jean Claude Latombe. Navigation strategies for exploring indoor environments. In *Int. J. Rob. Res.*, 2002.

[47] Cyrill Stachniss, Giorgio Grisetti, and Wolfram Burgard. Information gain-based exploration using rao-blackwellized particle filters. In *Robot. Sci. Syst. I*, volume 1, pages 65–72, 2005.

[48] Andreas Bircher, Mina Kamel, Kostas Alexis, Helen Oleynikova, and Roland Siegwart. Receding horizon path planning for 3D exploration and surface inspection. *Auton. Robots*, 42(2):291–306, 2018.

[49] Cindy Leung, Shoudong Huang, Ngai Kwok, and Gamini Dissanayake. Planning under uncertainty using model predictive control for information gathering. *Rob. Auton. Syst.*, 54(11):898–910, 2006.

[50] Allison Ryan and J. Karl Hedrick. Particle filter based information-theoretic active sensing. *Rob. Auton. Syst.*, 58(5):574–584, 2010.

[51] Yongyong Wei and Rong Zheng. Informative path planning for mobile sensing with reinforcement learning. In *IEEE Conference on Computer Communications*, pages 864–873, 2020.

[52] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents).* The MIT Press, 2005.

[53] Brian J. Julian, Sertac Karaman, and Daniela Rus. On mutual information-based control of range sensing robots for mapping applications. *Int. J. Rob. Res.*, 33(10):1375–1392, sep 2014.

[54] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* Wiley, 2005.

[55] Steven M LaValle. *Planning algorithms.* Cambridge university press, 2006.

[56] Mark Pfeiffer, Giuseppe Paolo, Hannes Sommer, Juan Nieto, Rol Siegwart, and Cesar Cadena. A data-driven model for interaction-aware pedestrian motion prediction in object cluttered environments. In *2018 IEEE Int. Conf. Robot. Autom.*, pages 1–8, 2018.

[57] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997.

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

[59] Yoshua Bengio, Jérome Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ACM Int. Conf. Proceeding Ser.*, volume 382, 2009.

[60] Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu. Stable baselines. `https://github.com/hill-a/stable-baselines`, 2018.

[61] Alexander Domahidi and Juan Jerez. FORCES professional. Embotech AG, `https://embotech.com/FORCES-Pro`.

[62] Michael Everett, Yu Fan Chen, and Jonathan P. How. Motion Planning Among Dynamic, Decision-Making Agents with Deep Reinforcement Learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[63] Arjun Majumdar, Gunjan Aggarwal, Bhavika Suresh Devnani, Judy Hoffman, and Dhruv Batra. ZSON: Zero-Shot Object-Goal Navigation using Multimodal Goal Embeddings. In *Advances in Neural Information Processing Systems*, 2022.

[64] Fei Xia, Amir R. Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson Env: Real-World Perception for Embodied Agents. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9068–9079, 2018.

[65] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D Data in Indoor Environments. In *2017 International Conference on 3D Vision (3DV)*, pages 667–676, 2017.

[66] Boyu Zhou, Yichen Zhang, Xinyi Chen, and Shaojie Shen. FUEL: Fast UAV Exploration Using Incremental Frontier Structure and Hierarchical Planning. *IEEE Robotics and Automation Letters*, 6(2):779–786, 2021.

[67] Josh Abramson, Arun Ahuja, Federico Carnevale, Petko Georgiev, Alex Goldin, Alden Hung, Jessica Landon, Jirka Lhotka, Timothy Lillicrap, Alistair Muldal, George Powell, Adam Santoro, Guy Scully, Sanjana Srivastava, Tamara von Glehn, Greg Wayne, Nathaniel Wong, Chen Yan, and Rui Zhu. Improving Multimodal Interactive Agents with Reinforcement Learning from Human Feedback, 2022.

[68] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, and Johnny Lee. Transporter Networks: Rearranging the Visual World for Robotic Manipulation. In *Proceedings of the 2020 Conference on Robot Learning*, 2021.

[69] Jonathan Spencer, Sanjiban Choudhury, Matthew Barnes, Matthew Schmittle, Mung Chiang, Peter Ramadge, and Sidd Srinivasa. Expert Intervention Learning. *Autonomous Robots*, 46(1):99–113, 2022.

[70] Zehui Meng, Hailong Qin, Ziyue Chen, Xudong Chen, Hao Sun, Feng Lin, and Marcelo H. Ang. A Two-Stage Optimized Next-View Planning Framework for 3-D Unknown Environment Exploration, and Structural Reconstruction. *IEEE Robotics and Automation Letters*, 2(3):1680–1687, 2017.

[71] Max Lodel, Bruno Brito, Álvaro Serra-Gómez, Laura Ferranti, Robert Babuška, and Javier Alonso-Mora. Where to Look Next: Learning Viewpoint Recommendations for Informative Trajectory Planning. In *2022 International Conference on Robotics and Automation (ICRA)*, 2022.

[72] B. Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97*, 1997.

[73] Nathan Hughes, Yun Chang, and Luca Carlone. Hydra: A Real-time Spatial Perception System for 3D Scene Graph Construction and Optimization. In *Robotics: Science and Systems XVIII*, 2022.

[74] Dorsa Sadigh, Anca Dragan, Shankar Sastry, and Sanjit Seshia. Active Preference-Based Learning of Reward Functions. In *Robotics: Science and Systems XIII*, 2017.

[75] Ralph Allan Bradley and Milton E. Terry. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika*, 39(3/4), 1952.

[76] Avrim Blum, Prasad Chalasani, Don Coppersmith, Bill Pulleyblank, Prabhakar Raghavan, and Madhu Sudan. The minimum latency problem. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing - STOC '94*, 1994.

[77] David Pisinger and Stefan Ropke. Large Neighborhood Search. In Michel Gendreau and Jean-Yves Potvin, editors, *Handbook of Metaheuristics*. 2019.

[78] Daniel J. Rosenkrantz, Richard E. Stearns, and Philip M. Lewis, II. An Analysis of Several Heuristics for the Traveling Salesman Problem. *SIAM Journal on Computing*, 6(3), 1977.

[79] G. A. Croes. A Method for Solving Traveling-Salesman Problems. *Operations Research*, 6(6), 1958.

[80] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2), 1968.

[81] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Proc-THOR: Large-Scale Embodied AI Using Procedural Generation. In *Advances in Neural Information Processing Systems*, 2022.

[82] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[83] 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, 2018.

[84] Devendra Singh Chaplot, Helen Jiang, Saurabh Gupta, and Abhinav Gupta. Semantic Curiosity for Active Visual Learning, June 2020. arXiv:2006.09367 [cs].

[85] Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. VLFM: Vision-Language Frontier Maps for Zero-Shot Semantic Navigation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[86] 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, May 2023. arXiv:2305.16925 [cs].

[87] 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, 2022.

[88] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual Language Maps for Robot Navigation, 2022.

[89] Muhammad Fadhil Ginting, Sung-Kyun Kim, David D. Fan, Matteo Palieri, Mykel J. Kochenderfer, and Ali-akbar Agha-Mohammadi. SEEK: Semantic Reasoning for Object Goal Navigation in Real World Inspection Tasks, May 2024. arXiv:2405.09822 [cs].

[90] Andreas Bircher, Mina Kamel, Kostas Alexis, Helen Oleynikova, and Roland Siegwart. Receding Horizon "Next-Best-View" Planner for 3D Exploration. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1462–1468, May 2016.

[91] Mihir Dharmadhikari, Tung Dang, Lukas Solanka, Johannes Loje, Huan Nguyen, Nikhil Khedekar, and Kostas Alexis. Motion Primitives-based Path Planning for Fast and Agile Exploration using Aerial Robots. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[92] Lukas Schmid, Michael Pantic, Raghav Khanna, Lionel Ott, Roland Siegwart, and Juan Nieto. An Efficient Sampling-Based Method for Online Informative Path Planning in Unknown Environments. *IEEE Robotics and Automation Letters*, 5:1–1, January 2020.

[93] Tung Dang, Christos Papachristos, and Kostas Alexis. Visual Saliency-Aware Receding Horizon Autonomous Exploration with Application to Aerial Robotics. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2526–2533, Brisbane, QLD, May 2018. IEEE.

[94] Titus Cieslewski, Elia Kaufmann, and Davide Scaramuzza. Rapid exploration with multi-rotors: A frontier selection method for high speed flight. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2135–2142, September 2017. ISSN: 2153-0866.

[95] Junlong Huang, Boyu Zhou, Zhengping Fan, Yilin Zhu, Yingrui Jie, Longwei Li, and Hui Cheng. FAEL: Fast Autonomous Exploration for Large-scale Environments With a Mobile Robot. *IEEE Robotics and Automation Letters*, 8(3):1667–1674, March 2023.

[96] Yichen Zhang, Xinyi Chen, Chen Feng, Boyu Zhou, and Shaojie Shen. FALCON: Fast Autonomous Aerial Exploration Using Coverage Path Guidance. *IEEE Transactions on Robotics*, 2024.

[97] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

[98] 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 *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[99] Daniel Mellinger and Vijay Kumar. Minimum snap trajectory generation and control for quadrotors. In *2011 IEEE international conference on robotics and automation*, pages 2520–2525. IEEE, 2011.

[100] Luxin Han, Fei Gao, Boyu Zhou, and Shaojie Shen. Fiesta: Fast incremental euclidean distance fields for online motion planning of aerial robots. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4423–4430. IEEE, 2019.

[101] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[102] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[103] Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023.

[104] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European conference on computer vision (ECCV)*, pages 85–100, 2018.

[105] Siyuan Wu. Risk-aware decentralized multi-mav planning in unknown and dynamic environments. *MSc. Thesis, Delft University of Technology*, 2023.

[106] 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, August 2020. arXiv:2006.13171 [cs].

[107] Naoki Yokoyama, Sehoon Ha, and Dhruv Batra. Success weighted by completion time: A dynamics-aware evaluation criteria for embodied navigation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1562–1569. IEEE, 2021.

[108] Boyu Zhou, Hao Xu, and Shaojie Shen. Racer: Rapid collaborative exploration with a decentralized multi-uav system. *IEEE Transactions on Robotics*, 39(3):1816–1835, 2023.

[109] Jingtian Yan, Xingqiao Lin, Zhongqiang Ren, Shiqi Zhao, Jieqiong Yu, Chao Cao, Peng Yin, Ji Zhang, and Sebastian Scherer. MUI-TARE: Cooperative Multi-Agent Exploration with Unknown Initial Position. *IEEE Robotics and Automation Letters*, 2023.

[110] Yulin Hui, Xuewei Zhang, Hongming Shen, Hanchen Lu, and Bailing Tian. DPPM: Decentralized Exploration Planning for Multi-UAV Systems Using Lightweight Information Structure. *IEEE Transactions on Intelligent Vehicles*, 2023.

[111] Nathan Hughes, Yun Chang, and Luca Carlone. Hydra: A real-time spatial perception system for 3d scene graph construction and optimization. *arXiv preprint arXiv:2201.13360*, 2022.

[112] Dengxin Dai and Luc Van Gool. Dark Model Adaptation: Semantic Image Segmentation from Daytime to Nighttime. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018.

[113] Angtian Wang, Yihong Sun, Adam Kortylewski, and Alan L. Yuille. Robust Object Detection Under Occlusion With Context-Aware CompositionalNets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[114] Mohamed Sayed and Gabriel Brostow. Improved Handling of Motion Blur in Online Object Detection. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[115] Antoni Rosinol, Andrew Violette, Marcus Abate, Nathan Hughes, Yun Chang, Jingnan Shi, Arjun Gupta, and Luca Carlone. Kimera: From SLAM to spatial perception with 3D dynamic scene graphs. *The International Journal of Robotics Research*, 40(12-14), 2021.

[116] Devendra Singh Chaplot, Helen Jiang, Saurabh Gupta, and Abhinav Gupta. Semantic Curiosity for Active Visual Learning. In *Computer Vision – ECCV 2020*, 2020.

[117] Julius Rückin, Federico Magistri, Cyrill Stachniss, and Marija Popović. Semi-Supervised Active Learning for Semantic Segmentation in Unknown Environments Using Informative Path Planning. *IEEE Robotics and Automation Letters*, 9(3), 2024.

[118] 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 *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[119] 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*, 9(5), 2024.

[120] Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. ConceptGraphs: Open-Vocabulary 3D Scene Graphs for Perception and Planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

# Glossary

| | |
|---|---|
| **CNN** | Convolutional Neural Network |
| **DRL** | Deep Reinforcement Learning |
| **FC** | Fully Connected (layer) |
| **IPP** | Informative Path Planning |
| **LLM** | Large Language Model |
| **LNS** | Large Neighborhood Search |
| **LSTM** | Long Short-Term Memory |
| **MAV** | Micro Aerial Vehicle |
| **MCTS** | Monte Carlo Tree Search |
| **MI** | Mutual Information |
| **MLP** | Minimum Latency Problem |
| **MPC** | Model Predictive Control |
| **OOI** | Object of Interest |
| **PLR** | Path Length Ratio |
| **PPO** | Proximal Policy Optimization |
| **RL** | Reinforcement Learning |
| **RNN** | Recurrent Neural Network |
| **SPL** | Success weighted by Path Length |
| **TSP** | Traveling Salesman Problem |
| **VLM** | Vision-Language Model |
| **WMLP** | Weighted Minimum Latency Problem |

# Curriculum Vitæ

## Max Lodel

24/06/1993 Born in Dresden, Germany.

## Education

2012 Abitur (high school diploma) at Martin-Andersen-Nexö-Gymnasium, Dresden, Germany.

2012 - 2020 Diplom-Ingenieur (M.Sc.) in Automotive Engineering from Technische Universität Bergakademie Freiberg, Germany.

2018 - 2019 Erasmus Semester at TalTech University, Tallinn, Estonia.

2020 - 2025 Ph.D. in Robotics at Technische Universiteit Delft, Netherlands.

## Experience

11/2016 - 04/2017 Internship in ESP Functional Software Development, Bosch Engineering GmbH, Abstatt, Germany.

11/2017 - 07/2018 Student Research Assistant, Fraunhofer Institute for Transportation and Infrastructure Systems IVI, Dresden, Germany.

05/2019 - 11/2019 Graduation Project in Automated Driving, Audi AG, Ingolstadt, Germany.

02/2025 - present Robotics Engineer at Sensmore, Berlin, Germany.

# List of Publications

## Referred Conference Proceedings

1. **M. Lodel**, B. Brito, A. Serra-Gomez, L. Ferranti, R. Babuška, J. Alonso-Mora, ”Where to Look Next: Learning Viewpoint Recommendations for Informative Trajectory Planning”, 2022 IEEE International Conference on Robotics and Automation (ICRA), Philadelphia, US, 2022.

## Under Review

1. **M. Lodel**, N. Wilde, R. Babuska, J. Alonso-Mora, ”Learning Semantic Priorities for Robotic Target Search Planning”.

2. N. Sethi*, **M. Lodel***, L. Ferranti, R. Babuska, J. Alonso-Mora, ”STEM: Semantic Target Search and Exploration using MAVs in Cluttered Environments”.

* Equal contribution.