Language-Enhanced Mobile Manipulation Framework for Efficient Object Search in Indoor Environments

Liding Zhang[1*], Zeqi Li[1*], Kuanqi Cai[1*], Qian Huang[1], Zhenshan Bing[2,1], Alois Knoll[1]

***Abstract*— Efficient object search in complex environments is critical for applications like household assistance, but traditional methods lack the contextual reasoning for unfamiliar settings. To address this, we propose the Goal-Oriented Dynamically Heuristic-Guided Hierarchical Search (GODHS) framework. GODHS leverages large language models (LLMs) to infer scene semantics and guide the robot through a multi-level decision hierarchy, mimicking human-like search strategies. We ensure the reliability of the LLM's reasoning by using structured prompts at each stage of the hierarchy. For mobile manipulation, we introduce a heuristic-based motion planner combining polar angle sorting and distance prioritization to generate efficient exploration paths. Evaluations in Isaac Sim show that GODHS achieves higher search efficiency compared to conventional, non-semantic strategies. Website and Video are available at: https://drapandiger.github.io/GODHS.**

I. Introduction

When humans search for objects in unfamiliar environments, they rely on a hierarchical understanding of semantic information to rapidly localize potential object placements [1]. Inspired by this ability, our approach emulates the human strategy of leveraging semantic cues—e.g., "pillows often lie on beds"—to guide the search process. Rather than exhaustively scanning an entire space, humans focus on "carriers" (like beds or tables) that are most likely to hold a target object, significantly reducing the search effort. Mobile manipulation has made great strides in recent years [2]. However, most robots still rely on purely spatial exploration strategies and ignore semantic cues, which makes them inefficient in unfamiliar environments [3].

In addition, current navigation systems exhibit two major shortcomings. First, although they may incorporate basic semantic labels, they often fail to reason about the relationships between known objects and unknown targets. Second, while vision-language models (VLMs) can align language with image features, they often lack the broad commonsense knowledge that large language models (LLMs) possess. In principle, LLMs combined with robust semantic segmentation could provide deeper real-world contextual reasoning.

To address these gaps, this paper introduces the *Goal-Oriented Dynamically Heuristic-Guided Hierarchical Search* (**GODHS**) framework, an exploratory study inspired by human search behavior. Building on the notion that people mentally decompose a search task—from room, to carrier, to specific features—GODHS employs LLMs to orchestrate a five-level search hierarchy: *scene* → *room* → *carrier* → *feature* → *item*. As illustrated in Fig. 1, each level narrows the search scope through an LLM-driven reasoning process.

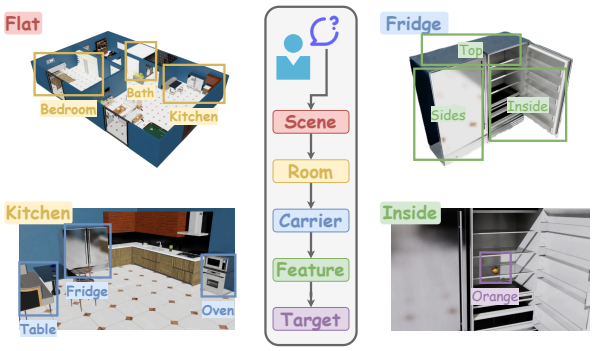

Fig. 1: The **GODHS framework**'s five-level hierarchy (Scene → Room → Carrier → Feature → Item). The example shows the LLM-guided process to find an ***orange*** by navigating the path: ***flat*** → ***kitchen*** → ***fridge*** → ***inside***.

Our work addresses two primary challenges. First, to achieve Reliable Hierarchical Reasoning, GODHS leverages LLMs with structured prompts and logical constraints to translate a high-level search goal into a multi-step, semantically-grounded plan. Second, for Efficient Physical Exploration, we propose a heuristic-based motion planner that combines polar and lexicographical sorting to efficiently generate and prioritize exploration poses, significantly reducing redundant travel and execution time.

II. Related Work

Robotic object search has progressed through several paradigms. Early geometric and probabilistic approaches established principles but were often limited to structured environments, as they required exhaustive object relationship priors [4], [5] and were confined to predefined taxonomies [6]. The advent of Vision-Language Models (VLMs) enabled open-vocabulary object recognition [7], but these methods often treat semantics as static attributes and can overfit to superficial correlations, struggling to chain contextual relationships for multi-step reasoning [8].

Recent integrations of Large Language Models (LLMs) can generate plausible search hypotheses [9], but many employ flat decision structures that inefficiently evaluate all object relationships [10]. While some methods capture object relationships more dynamically using scene graphs [11], a gap remains. Notably, while progress has been made in language grounding [12], few methods successfully emulate the human cognitive process of hierarchical task decomposition. Our work bridges this divide by synergizing LLM-driven commonsense reasoning with principled hierarchical action planning, avoiding both the inflexibility of purely geometric

[1]L. Zhang, Z. Li, K. Cai, Q. Huang, Z. Bing and A. Knoll are with the School of Computation, Information and Technology (CIT), Technical University of Munich, 80333 Munich, Germany. liding.zhang@tum.de

[2]Z. Bing is also with the State Key Laboratory for Novel Software Technology and the School of Science and Technology, Nanjing University (Suzhou Campus), China. *(Corresponding author: Zhenshan Bing.)*

*These authors contributed equally to this work.

The authors acknowledge the financial support by the Bavarian State Ministry for Economic Affairs, Regional Development and Energy (StMWi) for the Lighthouse Initiative KI.FABRIK (Phase 1: Infrastructure and the research and development program under grant no. DIK0249).

heuristics and the inefficiency of flat neural architectures.

III. Methodology

Our search approach follows a logical hierarchical progression, integrating environmental perception with commonsense reasoning from a large language model (LLM). The core of our method is the Goal-Oriented Dynamically Heuristic-Guided Hierarchical Search (GODHS) algorithm, detailed in Fig. 2 and Algorithm 1. GODHS is an efficient search approach guided by four key features: it is **Goal-Oriented**, pruning the search space at each level; **Dynamically Updated**, reordering priorities based on new information [13]; **Heuristic-Guided**, using LLM probabilities over strict optimization; and organized into a **Bounded Hierarchical** structure (Scene $\rightarrow$ Room $\rightarrow$ Carrier $\rightarrow$ Feature $\rightarrow$ Item) to avoid exhaustive searches [14].

A. GODHS Framework

The GODHS process begins with an exploration phase (Lines 2-7) to construct a global scene map $\mathcal{M}_{\mathcal{S}}$. The LLM infers room categories based on observed objects $\mathcal{O}$:

$$\mathbf{r}^* = \arg\max_{\mathsf{r} \in \mathcal{KB}} P\left(\mathsf{r} \mid \mathcal{O}\right), \tag{1}$$

and then ranks the rooms by the likelihood of containing the target t:

$$\mathbf{R} = \underset{\mathsf{r} \in \mathcal{R}}{\text{argsort}}\ P(\mathsf{r} \mid \mathcal{R}, \mathsf{t}). \tag{2}$$

The agent then navigates through rooms according to the prioritized list $\mathbf{R}$. In each room, it identifies plausible 'carriers' $\mathcal{C}$ for the target and ranks them similarly:

$$\mathbf{C} = \underset{\mathsf{c} \in \mathcal{C}}{\text{argsort}}\ P(\mathsf{c} \mid \mathcal{C}, \mathsf{t}). \tag{3}$$

For each carrier, the LLM determines a prioritized list of features (e.g., 'top', 'inside') to inspect, creating a final search plan. The robot then executes motion plans to visually inspect these features until the target is found (Lines 16-27).

To ensure the reliability of the LLM's guidance and address its potential for hallucination, we implement a multi-stage verification process for all LLM queries, inspired by structured reasoning techniques like Chain-of-Thought [15] and self-refinement. The cornerstone of this approach is a carefully structured prompt design. For any given task, such as determining the searchable features of a 'fridge' for an 'orange', the prompt explicitly defines the task, provides a set of valid outputs (e.g., 'top', 'bottom', 'sides', 'inside'), includes clarifying examples (e.g., a 'bathtub' corresponds to 'top'), and enforces a strict, machine-readable format. This structured prompting is crucial for grounding the LLM's abstract knowledge to the specific, operational needs of the robot at each level of the hierarchy, ensuring the output is semantically consistent and syntactically valid.

B. Heuristic-Based Pose Generation and Sorting

This subsection details our heuristic-based methodology for computing chassis (CH) and end-effector (EE) poses to efficiently generate structured search trajectories. The process, illustrated in Fig. 3, involves five stages: (1) extracting geometric features, (2) generating a set of EE poses $\mathcal{P}_{\mathcal{EE}}$, (3) selecting a corresponding set of CH poses $\mathcal{P}_{\mathcal{CH}}$, (4) validating CH-EE pairs with inverse kinematics (IK), and (5) sorting the resulting pose mappings for sequential execution.

The process begins with the carrier's point cloud ($\mathcal{M}_C$) and its 2D occupied grid ($\mathcal{M}_C^*$). From these, we extract four feature types. The **Top Surface** is defined as the set of points in $\mathcal{M}_C$ with the maximum z-value for each (x, y) coordinate in $\mathcal{M}_C^*$. The **Side Surfaces** are the vertical boundaries of the carrier, derived from the edges of $\mathcal{M}_C^*$. The **Bottom Area** is the footprint of the carrier at a fixed low height. The **Inside Area** is identified using a dedicated geometric analysis to find enclosed spaces.

Next, we generate the poses to visually cover these features. The set of EE poses, $\mathcal{P}_{\mathcal{EE}}$, is selected using a greedy algorithm [16] to ensure sufficient coverage. A corresponding set of CH poses, $\mathcal{P}_{\mathcal{CH}}$, is then generated to ensure a collision-free connection to all poses in $\mathcal{P}_{\mathcal{EE}}$. To guarantee feasibility, we solve the IK for each CH-EE pair using the Levenberg–Marquardt algorithm [17]; valid mappings are added to a dictionary $\mathcal{P}_{\mathcal{CH}}^{\mathcal{EE}}$.

Algorithm 1: ObjectSearchGODHS(s, t)

```
   Input  : s — scene name, t — target name
   Output: τ — found target
1  τ ← False, M_S, M_R, M_C ← ∅, R, C, F ← []
2  while not IsSceneMapComplete(M_S) do
3      EnterRandomRoom()
4      M_R ← LidarToMap(LidarData())
5      M_S ← UpdateSceneMap(M_S, M_R)
6      R, I_R ← RoomMap(M_R,
         InferRoom(SemSeg(CameraData())))
7  R ← SortRooms(R, t)
8  foreach r ∈ R do
9      MoveToRoom(r, M_R, I_R)
10     C ← ClassifyCarrier(SemSeg(CameraObservation()))
11     M_C, I_C ← GetCarrierPCL(C, CarrierObservation())
12     C ← SortCarriers(C, t)
13     foreach c ∈ C do
14         F ← {'top', 'bottom', 'sides', 'inside'}
15         F ← ReasonFeatures(t, F)
16         foreach f ∈ F do
17             M_F ← PredictFeatureMap(M_C, I_C, f)
18             P_EE ← DetermineEEPoses(M_F)
19             P_CH ← DetermineCHPoses(P_EE, M_F, I_F)
20             P_CH^EE ← CHToEEPoses(P_EE, P_CH, M_F, I_F)
21             P_CH^EE ← PosesSorting(P_CH^EE)
22             foreach P_CH ∈ P_CH do
23                 NavigateToCHPose(P_CH)
24                 foreach P_EE ∈ P_EE do
25                     NavigateToEEPose(P_EE)
26                     if t ∈ SemSeg(CameraData()) then
27                         return True
28 return False
```

Finally, to create an efficient path, the unsorted dictionary $\mathcal{P}_{\mathcal{CH}}^{\mathcal{EE}}$ is ordered to produce the final trajectory $\mathcal{P}_{\text{CH}}^{\text{EE}}$. For each pose in $\mathcal{P}_{\mathcal{CH}}$, the corresponding EE poses are sorted **lexicographically** to produce a systematic scanning motion. To ensure the robot moves efficiently around the carrier, the CH poses in $\mathcal{P}_{\mathcal{CH}}$ are sorted using centroid-aligned **Polar Angle Sorting** [18]. This is achieved by computing the geometric centroid $(\bar{x}, \bar{y})$ of the carrier and then sorting each chassis pose $(x_{\text{CH}}, y_{\text{CH}})$ by its angle ρ:

$$\rho = \arctan_2\left(y_{\text{CH}} - \bar{y},\ x_{\text{CH}} - \bar{x}\right). \tag{4}$$

This method ensures a smooth, ordered traversal around the carrier, preventing inefficient movements that can arise from simple distance-based sorting.

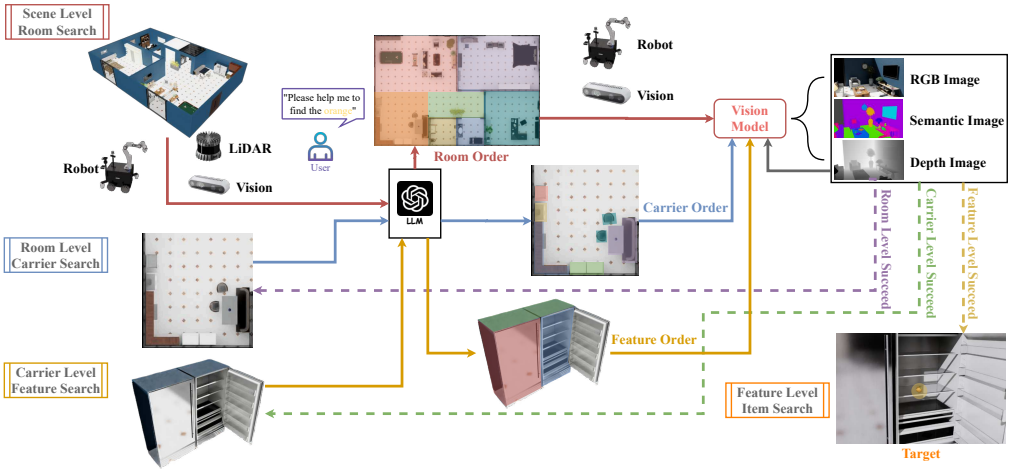

Fig. 2: **Complete Architecture of the GODHS System:** The process begins with a user's natural language query for a target object. The robot uses LiDAR and cameras to capture geometric and visual data, which an LLM processes to generate a topological map with semantic room labels. Guided by the LLM, the system prioritizes and navigates sequentially to the rooms most likely to contain the target. Within each room, the robot uses visual object detection to find candidate objects; the LLM then identifies which are plausible 'carriers' and ranks them. For each carrier, the LLM devises a search strategy by prioritizing specific features (e.g., top, interior). The loop terminates when the target is found and its identity is verified by the LLM through visual-language grounding.

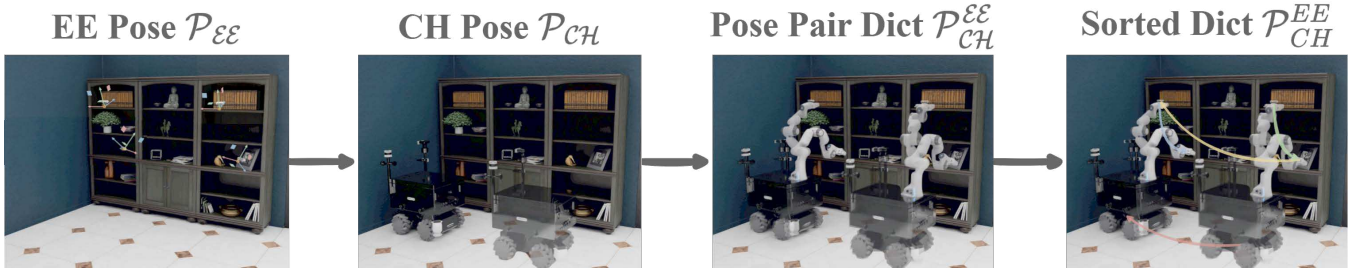

Fig. 3: **The pose sorting pipeline (left to right):** determining EE poses from carrier geometry, generating corresponding CH poses, verifying the pairs with IK, and prioritizing them using Polar Angle Sorting (for CH) and Lexicographical Sorting (for EE).

IV. Experimental Setup and Results

A. Evaluation system setup

The simulation framework uses NVIDIA Isaac Sim, offering photorealistic sensor data and accurate physics simulation. The robotic platform is DARKO, featuring an omnidirectional RB-KAIROS base and a 7-DOF Franka Emika Panda arm, integrated with an Ouster OS1 LiDAR and an Intel RealSense D435 camera. The system runs on ROS Noetic, employing the standard Navigation Stack for base control and MoveIt for manipulator planning. A local Ollama server powers the cognitive layer for LLM interaction.

B. Experimental Results

We conducted experiments in a simulated environment to evaluate two key aspects of our system: the overall search efficiency of the **GODHS framework** and the performance of our **heuristic-based motion planner**.

First, to test the system's search and semantic planning capabilities, experiments were run in a simulated "flat" scene with seven functional zones (e.g., kitchen, living room). A representative task involved locating an "orange" placed inside a fridge. Following an initial exploration to map the scene, the LLM-guided robot successfully navigated between rooms, identified the correct "carrier" (the fridge), and located the target inside.

To quantify this, we compared our method against two non-semantic baselines (Coverage and Random Walk) using several search efficiency metrics: Room Search Rate (R_r), Carrier Search Rate (R_c), and Item Search Rate (R_i). A lower value indicates a more efficient search. The Overall Search Rate (OSR) is a weighted average representing the total search cost:

$$OSR = w_1 \cdot R_r + w_2 \cdot R_c + w_3 \cdot R_i, \tag{5}$$

where weights are assigned as $w_1 = 0.2$, $w_2 = 0.3$, and $w_3 = 0.5$. As shown in Table I, the results from 81 experiments validate the efficiency of the GODHS framework. Guided by both tested LLMs, our system demonstrates significantly lower search rates across all categories compared to the baselines, confirming that the hierarchical, semantic guidance reduces search cost.

TABLE I: Search efficiency of different strategies. Lower values indicate higher efficiency.

Method	R_r **(%)**	R_c **(%)**	R_i **(%)**	**OSR (%)**
GPT-4o	**21.43**	20.53	21.17	**21.03**
Qwen2.5	33.85	**19.91**	**19.74**	22.61
Coverage	58.57	61.71	60.56	60.51
Random	47.14	52.10	53.38	51.75

We evaluated our motion planner's pose sorting strategies. We compared four configurations: an unoptimized baseline, sorting only end-effector (EE) poses, sorting only chassis

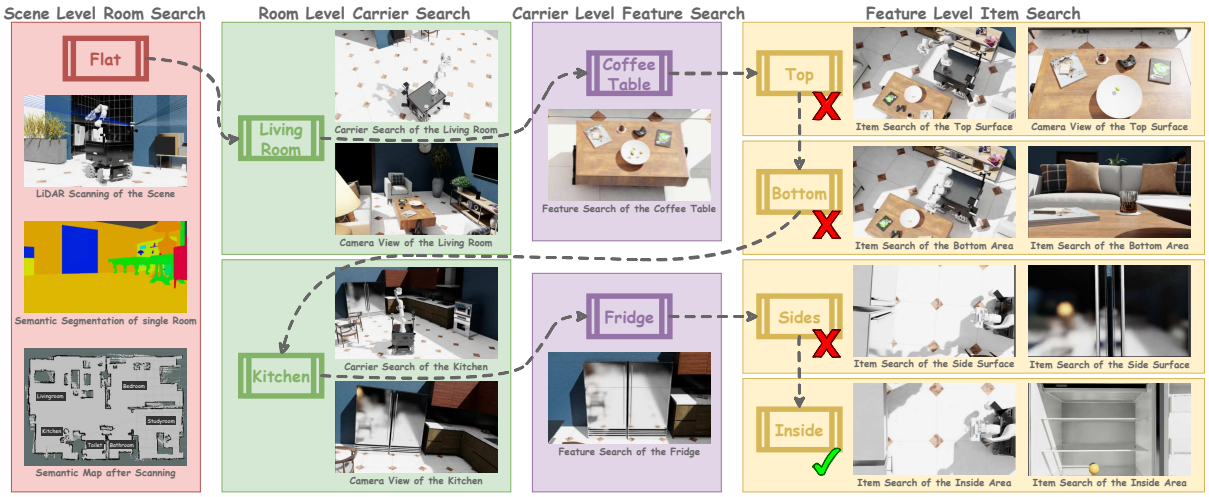

Fig. 4: Feasibility Example: Taking a Qwen2.5-7B-powered search as an example, the target is an orange. After an initial exploration to map the scene (lower left), the LLM guides the robot to the living room first. It inspects the coffee table's top and bottom surfaces without success. It then proceeds to the kitchen, inspects the side of the fridge, and finally opens the door to find the orange inside.

(CH) poses, and optimizing both together. As shown in Table II, applying lexicographical sorting to EE poses and polar angle sorting to CH poses independently yields significant improvements. When both strategies are combined, the system achieves the highest optimization in both path efficiency and overall execution time, confirming the effectiveness of our heuristic motion planning strategy.

TABLE II: Comparison of Sorting Methods for EE and CH Poses.

Sorting Method	EE Ratio	CH Ratio	Time Ratio
Unoptimized	2.81	2.37	1.00
EE Sorting	**1.75**	2.41	0.87
CH Sorting	2.79	**1.60**	0.83
Both Optimized	**1.77**	**1.59**	**0.66**

During testing, we also identified three categories of failure modes: (i) **hardware limitations**, such as the manipulator's reach; (ii) **insufficient common-sense in the LLM**, like attempting to inspect a non-openable surface; and (iii) **semantic ambiguity**, such as misclassifying objects.

V. Conclusion

In this work, we presented the **GODHS Framework**, which integrates an LLM's commonsense reasoning with a multi-level decision process to improve search efficiency. This is achieved by using structured prompts to ensure reliable reasoning and a heuristic-based motion planner with **pose sorting** to generate efficient exploration trajectories. Experiments conducted in simulation demonstrated the feasibility of our approach and more efficient search performance compared to non-semantic strategies. Future work will focus on deploying the framework on a physical robot and exploring the integration of adaptively prolated optimal models [19] and social navigation models [20] to enhance its capabilities.

References

[1] L. Zhang, K. Cai, Z. Sun, Z. Bing, C. Wang, L. Figueredo, S. Haddadin, and A. Knoll, "Motion planning for robotics: A review for sampling-based planners," *Biomimetic Intelligence and Robotics*, vol. 5, no. 1, p. 100207, 2025.

[2] B. Kuipers, "The spatial semantic hierarchy," *Artificial intelligence*, vol. 119, no. 1-2, pp. 191–233, 2000.

[3] L. Zhang, K. Cai, Y. Zhang, Z. Bing, C. Wang, F. Wu, S. Haddadin, and A. Knoll, "Estimated informed anytime search for sampling-based planning via adaptive sampler," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 18 580–18 593, 2025.

[4] L. E. Wixson and D. H. Ballard, "Using intermediate objects to improve the efficiency of visual search," *Int. J. Comput. Vision*, vol. 12, no. 2–3, p. 209–230, Apr. 1994.

[5] E. Gelenbe and Y. Cao, "Autonomous search for mines," *European Journal of Operational Research*, vol. 108, no. 2, pp. 319–333, 1998.

[6] J. K. Li, D. Hsu, and W. S. Lee, "Act to see and see to act: Pomdp planning for objects search in clutter," pp. 5701–5707, 2016.

[7] S. Patki, E. Fahnestock, T. M. Howard, and M. R. Walter, "Language-guided semantic mapping and mobile manipulation in partially observable environments," 2019.

[8] Y. Tang, M. Wang, Y. Deng, Z. Zheng, J. Deng, and Y. Yue, "Openin: Open-vocabulary instance-oriented navigation in dynamic domestic environments," 2025.

[9] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, and Y. Su, "Llm-planner: Few-shot grounded planning for embodied agents with large language models," 2023.

[10] L. L. Wong, L. P. Kaelbling, and T. Pérez, "Manipulation-based active search for occluded objects," IEEE, pp. 2814–2819, 2013.

[11] D. Honerkamp, M. Büchner, F. Despinoy, T. Welschehold, and A. Valada, "Language-grounded dynamic scene graphs for interactive object search with mobile manipulation," *IEEE Robotics and Automation Letters*, vol. 9, no. 10, p. 8298–8305, Oct. 2024.

[12] M. R. Walter, S. Patki, A. F. Daniele, E. Fahnestock, F. Duvallet, S. Hemachandra, J. Oh, A. Stentz, N. Roy, and T. M. Howard, "Language understanding for field and service robots in a priori unknown environments," 2021.

[13] D. D. Sleator and R. Endre Tarjan, "A data structure for dynamic trees," *Journal of Computer and System Sciences*, vol. 26, no. 3, pp. 362–391, 1983.

[14] A. Aydemir, A. Pronobis, M. Göbelbecker, and P. Jensfelt, "Active visual object search in unknown environments using uncertain semantics," *IEEE Transactions on Robotics*, vol. 29, no. 4, pp. 986–1002, 2013.

[15] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," 2023.

[16] E. DIJKSTRA, "A note on two problems in connexion with graphs." pp. 269–271, 1959.

[17] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," 1986.

[18] R. L. Graham, "An efficient algorithm for determining the convex hull of a finite planar set," *Inf. Process. Lett.*, vol. 1, pp. 132–133, 1972.

[19] L. Zhang, S. Wang, K. Cai, Z. Bing, F. Wu, C. Wang, S. Haddadin, and A. Knoll, "APT*: Asymptotically optimal motion planning via adaptively prolated elliptical r-nearest neighbors," *IEEE Robotics and Automation Letters*, vol. 10, no. 10, pp. 10 242–10 249, 2025.

[20] K. Cai, W. Chen, C. Wang, H. Zhang, and M. Q.-H. Meng, "Curiosity-based robot navigation under uncertainty in crowded environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 800–807, 2023.