Language-Enhanced Mobile Manipulation 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*— Enabling robots to efficiently search for and identify objects in complex, unstructured environments is critical for diverse applications ranging from household assistance to industrial automation. However, traditional scene representations typically capture only static semantics and lack interpretable contextual reasoning, limiting their ability to guide object search in completely unfamiliar settings. To address this challenge, we propose a language-enhanced hierarchical navigation framework that tightly integrates semantic perception and spatial reasoning. Our method, Goal-Oriented Dynamically Heuristic-Guided Hierarchical Search (GODHS), leverages large language models (LLMs) to infer scene semantics and guide the search process through a multi-level decision hierarchy. Reliability in reasoning is achieved through the use of structured prompts and logical constraints applied at each stage of the hierarchy. For the specific challenges of mobile manipulation, we introduce a heuristic-based motion planner that combines polar angle sorting with distance prioritization to efficiently generate exploration paths. Comprehensive evaluations in Isaac Sim demonstrate the feasibility of our framework, showing that GODHS can locate target objects with higher search efficiency compared to conventional, non-semantic search 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.

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.

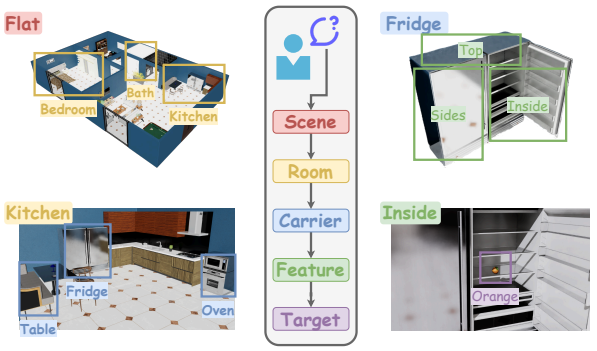

Fig. 1: The **GODHS framework** divides each scene into five strict levels: Scene, Room, Carrier, Feature, and Item. For instance, after entering the ***flat***, the robot uses mapping and semantic segmentation to determine and prioritize the types of rooms through a LLM. For the first room ***kitchen***, the LLM classifies and prioritizes carriers such as furniture or devices. For the first carrier ***fridge***, the LLM identifies and prioritizes the feature regions worth searching. Finally, for the first feature ***inside***, the robot searches for the presence of the target object ***orange***.

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* $\rightarrow$ *room* $\rightarrow$ *carrier* $\rightarrow$ *feature* $\rightarrow$ *item*. As illustrated in Fig. 1, each level narrows the search scope through an LLM-driven reasoning process.

Implementing this hierarchical pipeline presents two primary challenges that our work addresses:

- **Reliable Hierarchical Reasoning:** The system must translate a high-level search goal into a multi-step, semantically-grounded plan. GODHS achieves this by leveraging LLMs to hierarchically decompose the problem. To ensure the reliability of the LLM's guidance, we employ structured prompts and logical constraints at each level of the hierarchy.
- **Efficient Physical Exploration:** A mobile manipulator exploring a carrier faces a vast action space for positioning its base and end-effector. 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 three main paradigms. Early geometric approaches established principles but faced critical limitations. These methods constrained

[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).

search regions using spatial affordances [3] or modeled uncertainty with probabilistic frameworks [4]–[6]. However, they were often effective only in structured environments, as they required exhaustive object relationship priors and could not infer latent functional associations (e.g., that medicine belongs in a cabinet). Underpinning these systems are foundational motion planning algorithms [7], which have evolved from sampling-based approaches to adapted high-DoF robotic systems for real-time planning [8], [9]. Furthermore, traditional decision-making frameworks like POMDPs [10] remained confined to predefined taxonomies, limiting their adaptability to novel objects.

The advent of VLMs enabled open-vocabulary object recognition but introduced new constraints. While semantic mapping systems [11] enriched spatial maps with object labels, they often treated semantics as static attributes, overlooking dynamic relational reasoning. Other works used hierarchical graphs to model object interactions [12], but these rigid structures typically required complete environmental pre-knowledge. Modern VLM-integrated navigation methods [13], [14] can locate open-vocabulary targets but may overfit to superficial visual-text correlations [15], struggling to chain contextual relationships for multi-step reasoning.

Recent integrations of LLMs show strong potential but still face challenges. LLM-augmented planners generate plausible search hypotheses [16], yet many rely on flat decision structures that inefficiently evaluate object relationships. Scene-graph methods capture relationships more dynamically [17], [18] but often lack incremental refinement and assume high observability [19]. While progress has been made in language grounding, few methods emulate human-like 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 heuristics and the inefficiency of flat neural architectures.

III. Methodology

Our search approach follows a logical hierarchical progression to locate the ultimate target. We construct a hierarchically expandable algorithm, GODHS, that integrates environmental perception with commonsense reasoning from a large language model (LLM) to derive semantic action sequences(Sec. III-A). To ensure the reliability of the LLM-driven reasoning, we employ structured prompts and constraints at each stage of the hierarchy. To ensure the mobile robot advances sequentially around the carrier, we employ dictionary mapping and polar angle sorting (Sec. III-B).

A. GODHS Framework

Goal-Oriented Dynamically Heuristic-Guided Hierarchical Search is an efficient search approach that combines cognitive reasoning, sensory data processing, and decision-making strategies. The detailed process is shown in Figure 2. The GODHS algorithm is designed primarily to locate target objects in complex environments. It has four key features:

- **Goal-Oriented:** The search is consistently focused on the final target, allowing the system to effectively prune the search space at each level of the hierarchy.
- **Dynamically Updated:** The system dynamically reassesses and reorders search priorities based on new information gathered during execution, inspired by dynamic tree search structures [20].

Algorithm 1: ObjectSearchGODHS(s, t)

Input : s — scene name, t — target name
Output: τ — found target

1 $\tau \leftarrow$ **False**, $\mathcal{M}_\mathcal{S}, \mathcal{M}_\mathcal{R}, \mathcal{M}_\mathcal{C} \leftarrow \varnothing$, $\mathbf{R}, \mathbf{C}, \mathbf{F} \leftarrow [\,]$
2 EnterScene(s)
3 **while** *not IsSceneMapComplete($\mathcal{M}_\mathcal{S}$)* **do**
4 EnterRandomRoom()
5 $\mathcal{M}_\mathcal{R} \leftarrow$ LidarToMap(LidarData())
6 $\mathcal{M}_\mathcal{S} \leftarrow$ UpdateSceneMap($\mathcal{M}_\mathcal{S}$, $\mathcal{M}_\mathcal{R}$)
7 $\mathcal{R}, \mathcal{I}_\mathcal{R} \leftarrow$ RoomMap($\mathcal{M}_\mathcal{R}$, InferRoom(SemSeg(CameraData())))
8 $\mathbf{R} \leftarrow$ SortRooms($\mathcal{R}$, t)
9 **foreach** r $\in \mathbf{R}$ **do**
10 MoveToRoom(r, $\mathcal{M}_\mathcal{R}$, $\mathcal{I}_\mathcal{R}$)
11 $\mathcal{C} \leftarrow$ ClassifyCarrier(SemSeg(CameraObservation()))
12 $\mathcal{M}_\mathcal{C}, \mathcal{I}_\mathcal{C} \leftarrow$ GetCarrierPCL($\mathcal{C}$, CarrierObservation())
13 $\mathbf{C} \leftarrow$ SortCarriers($\mathcal{C}$, t)
14 **foreach** c $\in \mathbf{C}$ **do**
15 $\mathcal{F} \leftarrow$ {'top', 'bottom', 'sides', 'inside'}
16 $\mathbf{F} \leftarrow$ ReasonFeatures(t, $\mathcal{F}$)
17 **foreach** f $\in \mathbf{F}$ **do**
18 $\mathcal{M}_\mathcal{F} \leftarrow$ PredictFeatureMap($\mathcal{M}_\mathcal{C}$, $\mathcal{I}_\mathcal{C}$, f)
19 $\mathcal{P}_{\mathcal{E}\mathcal{E}} \leftarrow$ DetermineEEPoses($\mathcal{M}_\mathcal{F}$)
20 $\mathcal{P}_{\mathcal{C}\mathcal{H}} \leftarrow$ DetermineCHPoses($\mathcal{P}_{\mathcal{E}\mathcal{E}}$, $\mathcal{M}_\mathcal{F}$, $\mathcal{I}_\mathcal{F}$)
21 $\mathcal{P}^{\mathcal{E}\mathcal{E}}_{\mathcal{C}\mathcal{H}} \leftarrow$ CHToEEPoses($\mathcal{P}_{\mathcal{E}\mathcal{E}}$, $\mathcal{P}_{\mathcal{C}\mathcal{H}}$, $\mathcal{M}_\mathcal{F}$, $\mathcal{I}_\mathcal{F}$)
22 $\mathcal{P}^{\mathrm{EE}}_{\mathrm{CH}} \leftarrow$ PosesSorting($\mathcal{P}^{\mathcal{E}\mathcal{E}}_{\mathcal{C}\mathcal{H}}$)
23 **foreach** $\mathbf{P}_{CH} \in \mathcal{P}_{CH}$ **do**
24 NavigateToCHPose($\mathbf{P}_{\mathrm{CH}}$)
25 **foreach** $\mathbf{P}_{EE} \in \mathcal{P}_{EE}$ **do**
26 NavigateToEEPose($\mathbf{P}_{\mathrm{EE}}$)
27 **if** t $\in$ *SemSeg(CameraData())* **then**
28 **return True**
29 **return False**

- **Heuristic-Guided:** The process is guided by contextual probabilities from the LLM, which infers likely search locations based on commonsense knowledge rather than strict mathematical optimization.
- **Bounded Hierarchical:** The search space is organized into a five-level hierarchy (Scene $\rightarrow$ Room $\rightarrow$ Carrier $\rightarrow$ Feature $\rightarrow$ Item), progressively narrowing the scope to avoid an exhaustive global search [21].

The full process is detailed in Algorithm 1, which takes a scene name s and a target t as input. The algorithm initializes the required maps ($\mathcal{M}_\mathcal{S}, \mathcal{M}_\mathcal{R}, \mathcal{M}_\mathcal{C}$) and lists ($\mathbf{R}, \mathbf{C}, \mathbf{F}$). The process begins with an autonomous exploration phase (Lines 2-8), where the robot constructs a global scene map $\mathcal{M}_\mathcal{S}$ by sequentially visiting and mapping all accessible rooms. For each room, a local map $\mathcal{M}_\mathcal{R}$ is built using LiDAR data and integrated into the global map: $\mathcal{M}_S \leftarrow \mathcal{M}_S \cup (\mathcal{M}_R \setminus \mathcal{M}_S)$.

The function InferRoom$(\cdot)$ in Line 7 employs an LLM to predict the room category $\mathbf{r}^*$ based on the set of observed objects $\mathcal{O}$:

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

where $\mathcal{KB}$ represents the commonsense knowledge of the pre-trained LLM. This generates a room-to-map correspondence $\mathcal{I}_\mathcal{R} : \mathcal{R} \rightarrow \mathcal{M}_\mathcal{R}$. Finally, the language model ranks

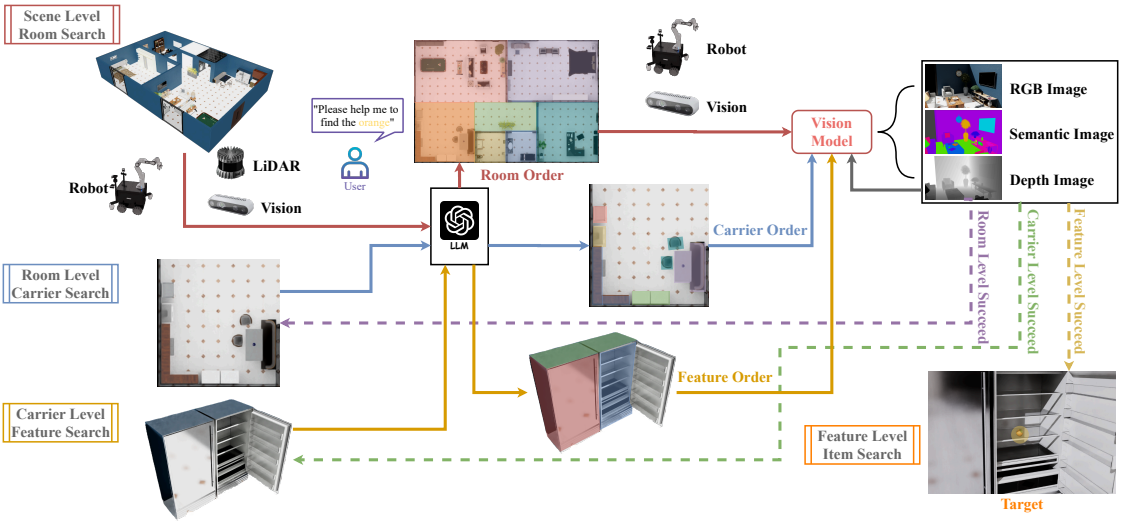

Fig. 2: **Complete Architecture of the GODHS System:** First, the user provides the target object through natural language input, which is then processed by a LLM to extract semantic intent. The robot captures geometric data using LiDAR and RGB images via cameras, generating a topological map with room names through the language model. Within the known map, the LLM-driven prioritization ranks rooms based on the likelihood of containing the target object and navigates to them sequentially. Within each room, the robot detects candidate objects through visual object detection, then utilizes the LLM to analyze whether they are carriers and ranks all collected carriers by the probability of containing the target. For each carrier, the LLM plans a hierarchical search strategy: first analyzing geometric features to identify subregions worth searching (top, interior, etc.), then prioritizing and searching these subregions according to probabilities assigned by the LLM. The loop terminates when the LLM verifies target recognition through visual-language grounding of sensor data.

elements of $\mathcal{R}$ by likelihood of target $\mathtt{t}$ in $\mathcal{R}$:

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

The agent navigates to rooms according to the order in $\mathbf{R}$. Within each room, the ClassifyCarrier$(\cdot)$ function in Line 11 prompts the LLM to identify which of the observed objects are plausible 'carriers' for the target $\mathtt{t}$. This step directly yields a filtered list of carrier objects $\mathcal{C} \subseteq \mathcal{O}$ without relying on an explicit numerical threshold. Focused scanning to capture carrier point clouds $\mathcal{M}_C$ is performed and carrier-to-map correspondence $\mathcal{I}_C : \mathcal{C} \to \mathcal{M}_C$ will be established. These identified carriers are then ranked by the LLM based on their relevance to the target $\mathtt{t}$:

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

Guided by the prioritized carrier list $\mathbf{C}$, the agent sequentially inspects each carrier $\mathtt{c} \in \mathbf{C}$. At the Feature Level, the agent must determine which specific parts of the carrier are most relevant for finding the target $\mathtt{t}$. We prompt the LLM to perform a direct selection and ranking task. Given a set of spatial regions $\mathcal{F} = \{$'top', 'bottom', 'sides', 'inside'$\}$, the LLM is tasked to return an ordered list of the most plausible features to inspect for the given carrier and target. This process yields a final, prioritized sequence of features $\mathbf{F} = \underset{\mathtt{f}\in\mathcal{F}}{\text{argsort}}\, P(\mathtt{f} \mid \mathcal{F}, \mathtt{t})$. The final stage of the algorithm 1 in Lines 17-28 then iterates through this robustly generated list $\mathbf{F}$, creating and executing motion plans to visually inspect each feature in order, until the target is found.

To ensure the reliability of the LLM's guidance within the GODHS framework, we must address the model's inherent tendency for statistical hallucination [22]. Inspired by recent structured reasoning techniques like Chain-of-Thought [23] and self-refinement [24], we implement a multi-stage verification process for LLM queries. This process, illustrated in Fig. 3 (Left), involves cleaning and structuring the input, executing the core reasoning task, and correcting the output to ensure it is semantically consistent and syntactically valid for the robot.

The cornerstone of this approach is a carefully structured prompt design, as shown in Fig. 3 (Right). For example, to determine the searchable features for a 'fridge' potentially containing an 'orange', the prompt explicitly defines the task, constrains the possible outputs to a predefined set ('top', 'bottom', 'sides', 'inside'), provides clarifying examples (e.g., a 'bathtub' should return 'top'), and enforces a strict, machine-readable output 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.

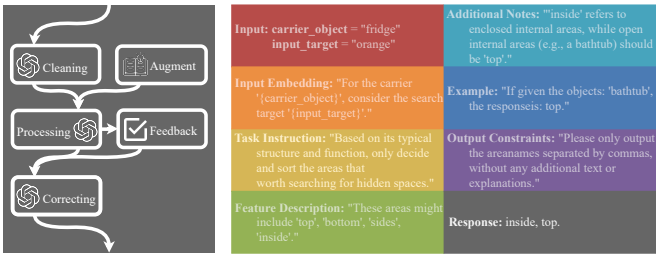

Fig. 3: **LLM Reasoning Process (Left):** A multi-stage process involving data cleaning, processing with optional knowledge augmentation, and correcting with optional feedback ensures reliable output. **Prompt Design (Right):** An example of a structured prompt used to ground the LLM's reasoning for a specific task.

B. Heuristic-Based Pose Generation and Sorting

This subsection details our heuristic-based methodology for computing chassis (CH) and end-effector (EE) poses. The goal is to efficiently generate structured search trajectories that enable the robot to systematically inspect a carrier's features while reducing redundant motion. The process involves

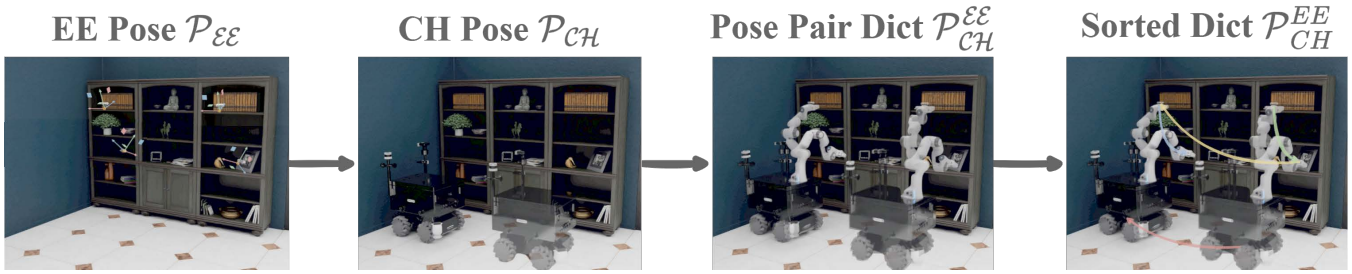

Fig. 4: **Leftmost:** Determine EE poses via carrier geometry analysis. **Second Left:** Generate CH poses through greedy EE pose exploration. **Second Right:** Verify CH-EE pairs with inverse kinematics validation. **Rightmost:** Prioritize CH poses by Polar Angle Sorting and EE poses by Lexicographical Sorting.

five sequential stages: (1) extraction of geometric features $\mathcal{M}_F$ from point cloud data, (2) generation and selection of EE poses $\mathcal{P}_{\mathcal{EE}}$, (3) fast evaluation and selection of chassis poses $\mathcal{P}_{\mathcal{CH}}$ via geometric solution, (4) validation of the relation between CH and EE $\mathcal{P}_{\mathcal{CH}}^{\mathcal{EE}}$ with subsequent inverse kinematics (IK) solving, and (5) sorting of the resulting pose mappings $\mathcal{P}_{\text{CH}}^{\text{EE}}$ for sequential execution.

The process begins with the acquisition of the point cloud of single carrier $\mathcal{M}_C \subset \mathbb{R}^3$ and the occupied grid of the carrier $\mathcal{M}_C^* \subset \mathbb{R}^2$. These data sources are used to extract the carrier's relevant features, collectively denoted as $\mathcal{M}_F$. The features are subdivided into four components: Top Surface, Side Surface, Bottom Area and Inside Area.

The top surface feature $\mathcal{M}_F^{\text{top}}$ is extracted directly from point cloud $\mathcal{M}_C$. For each (x_F, y_F) coordinate in the grid, the top surface point is identified as the point with the maximum z_F value. Formally, if we denote by $\mathcal{M}_C^*$ the grid cell corresponding to a particular (x_F, y_F) location, then

$$\mathcal{M}_F^{\text{top}} = \left\{ (x_F, y_F, z_F) \,\middle|\, \begin{array}{l} (x_F, y_F) \in \mathcal{M}_C^*, \\ (x_F, y_F, z_F') \in \mathcal{M}_C, \\ z_F = \\ \max\{z_F' \mid (x_F, y_F, z_F')\} \end{array} \right\}. \quad (4)$$

The side surface feature $\mathcal{M}_F^{\text{sides}}$ is derived from the boundary of the volumetric occupied grid $\mathcal{M}_C^*$. For each (x_F, y_F) coordinate lying on the topologically defined edge of $\mathcal{M}_C^*$ (denoted by $\partial\mathcal{M}_C^*$), the corresponding side surface value is determined by extracting the maximum z_F value in $\mathcal{M}_C$ over the range starting from $z_F = 0$. This could be expressed as:

$$\mathcal{M}_F^{\text{sides}} = \left\{ (x_F, y_F, z_F) \,\middle|\, \begin{array}{l} (x_F, y_F) \in \partial\mathcal{M}_C^*, \\ (x_F, y_F, z_F') \in \mathcal{M}_C, \\ z_F \in \\ [0, \max\{z_F' \mid (x_F, y_F, z_F')\}] \end{array} \right\}. \quad (5)$$

This representation captures the vertical boundaries of the carrier. The bottom area feature $\mathcal{M}_F^{\text{bottom}}$ is defined by the intersection of the occupied grid $\mathcal{M}_C^*$ with a spatially constrained horizontal plane at a fixed height $z_F = z_{F0}$:

$$\mathcal{M}_F^{\text{bottom}} = \left\{ (x_F, y_F, z_{F0}) \,\middle|\, (x_F, y_F) \in \mathcal{M}_C^* \right\}. \quad (6)$$

The inside feature is identified using a dedicated geometric analysis procedure designed to detect potential openings or enclosed spaces on the carrier.

For the mobile manipulator, two types of poses are essential: the chassis pose and the EE pose. The candidate chassis pose is defined as $\mathbf{p}_{CH}^{Can} = (x_{CH}, y_{CH}, \theta_{CH})$, where (x_{CH}, y_{CH}) represents the base position and θ_{CH} the yaw angle. The candidate EE pose is defined as $\mathbf{p}_{EE}^{Can} = (x_{EE}, y_{EE}, z_{EE}, \phi_{EE}, \theta_{EE}, \psi_{EE})$. The direction vector of the camera's view can be expressed as $\mathbf{v}_{\text{dir}} = (\cos(\theta_{EE}) \cdot \cos(\psi_{EE}), \cos(\theta_{EE}) \cdot \sin(\psi_{EE}), \sin(\theta_{EE}))$. The vector pointing from the camera to the surface point is $\mathbf{v}_{\text{point}} = (x_F - x_{EE}, y_F - y_{EE}, z_F - z_{EE})$. The surface point is considered covered by the vision cone of the camera, if the conditions of horizontal and vertical angle $\theta_{\text{horizontal}} = \arccos\left(\frac{v_{\text{dir},x} \cdot v_{\text{point},x} + v_{\text{dir},z} \cdot v_{\text{point},z}}{\|v_{\text{dir}}\| \|v_{\text{point}}\|}\right) < \text{FoV}_{\text{horizontal}}/2$ and $\theta_{\text{vertical}} = \arccos\left(\frac{v_{\text{dir},y} \cdot v_{\text{point},y} + v_{\text{dir},z} \cdot v_{\text{point},z}}{\|v_{\text{dir}}\| \|v_{\text{point}}\|}\right) < \text{FoV}_{\text{vertical}}/2$ are satisfied, and the candidate pose earns one point. We employ a greedy algorithm [25] as a heuristic to select a set of EE poses $\mathcal{P}_{\mathcal{EE}}$ that provide sufficient visual coverage.

The set of chassis poses $\mathcal{P}_{\mathcal{CH}}$ is selected from $\mathbf{p}_{CH}^{Can}$ via deterministic greedy algorithm based on $\mathcal{P}_{\mathcal{EE}}$. If a geometric solution can establish a collision-free connection, one point is awarded. The selection continues until $\mathcal{P}_{\mathcal{CH}}$ is found that spatially covers all elements in $\mathcal{P}_{\mathcal{EE}}$.

Since fast-solving cannot guarantee a feasible solution for the actual robot operation, and directly solving the IK for all poses would lead to excessive computational overhead, we perform IK solving with Levenberg–Marquardt algorithm [26] for each chassis pose $\mathbf{P}_{\text{CH}}$ in $\mathcal{P}_{\mathcal{CH}}$ and each end-effector pose $\mathbf{P}_{\text{EE}}$ in $\mathcal{P}_{\mathcal{EE}}$. If numerically stable solutions exist, the mapping is added to the dictionary $\mathcal{P}_{\mathcal{CH}}^{\mathcal{EE}}$.

The dictionary $\mathcal{P}_{\mathcal{CH}}^{\mathcal{EE}}$ is non-sequential, so a sorting method is required to obtain an ordered version $\mathcal{P}_{\text{CH}}^{\text{EE}}$. For each end-effector pose corresponding to a chassis pose, we apply Lexicographical Sorting based on the sequence $z_{EE}, y_{EE}, x_{EE}, \psi_{EE}, \theta_{EE}, \phi_{EE}$.

For all chassis poses, we do not simply sort based on spatial distance but instead ensure movement proceeds clockwise around the carrier. To achieve this, we use centroid-aligned Polar Angle Sorting [27], and the geometric centroid $(\bar{x}, \bar{y}) = \left(\frac{1}{n}\sum_{i=1}^{n} x_i, \frac{1}{n}\sum_{i=1}^{n} y_i\right)$ is computed where $(x_i, y_i) \in \mathcal{M}_C$, with n being the size of the point set. Then we can calculate its angle relative to the average point by:

$$\rho = \arctan_2\left(y_{\text{CH}} - \bar{y},\, x_{\text{CH}} - \bar{x}\right), \quad (7)$$

where $(x_{\text{CH}}, y_{\text{CH}})$ is the position of each chassis. Following the angle ρ, the chassis poses will be sorted. Using this method, the movement platform can advance around the carrier while steadily maintaining its direction of movement, avoiding the situation where sorting based on the nearest spatial distance might result in the actual path distance being significantly greater than the spatial distance.

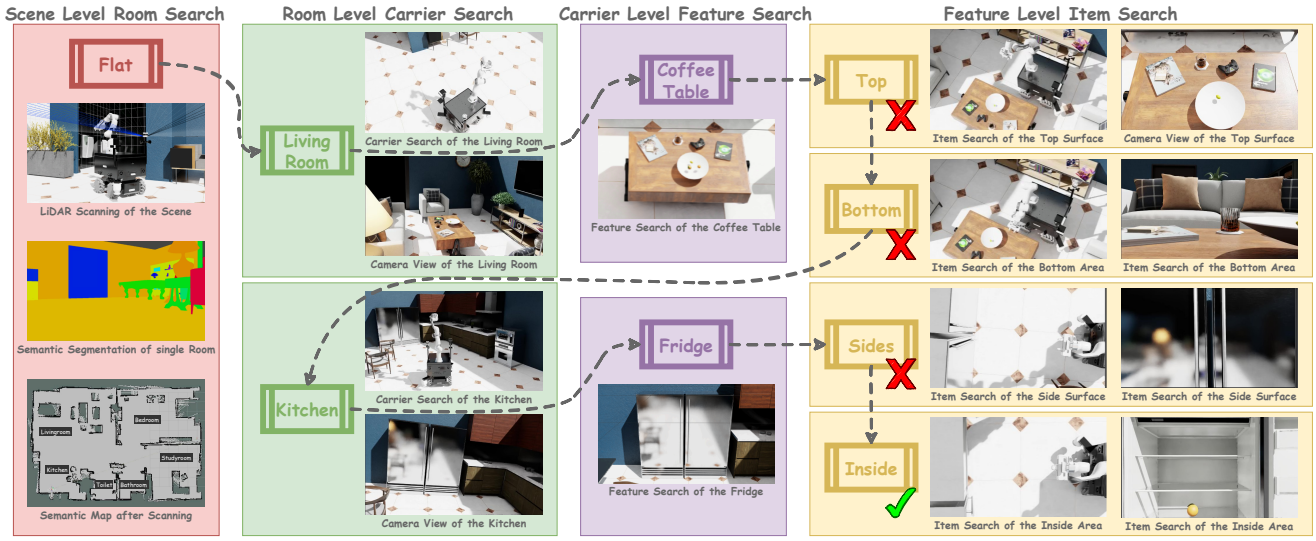

Fig. 5: **Feasibility Example:** Taking a Qwen2.5-7B-powered search as an example, the target is an orange. After an initial exploration to map the scene (lower left), the LLM guides the robot to the living room first. It inspects the coffee table's top and bottom surfaces without success. It then proceeds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

IV. Experimental Setup and Results

A. Evaluation system setup

The simulation framework is developed using 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-based interaction.

B. Feasibility Testing

The feasibility test evaluates the system's target search and semantic planning capabilities in a constrained indoor environment. Experiments were conducted in a simulated "flat" scene with seven functional zones (e.g., kitchen, living room, bedroom), as shown in Fig. 5. A representative task involves locating an "orange" placed inside a fridge, requiring the robot to navigate across rooms and identify relevant carriers and features. The process begins with environment exploration to build a complete occupancy map. Semantic segmentation detects objects, which are then processed by the LLM to infer room categories (e.g., recognizing a bed and dressing table suggests a bedroom). Based on the query "orange," the LLM prioritizes searching the living room followed by the kitchen. In the living room, the robot inspects the coffee table's top and bottom but finds nothing. It then moves to the kitchen, identifies the fridge as the most likely carrier, and successfully locates the orange inside.

To evaluate the search efficiency of our approach, we conducted 81 experiments using Qwen2.5-7B and GPT-4o. We define search efficiency metrics: the Room Search Rate (R_r), Carrier Search Rate (R_c), and Item Search Rate (R_i), which represent the percentage of rooms, carriers, or items, respectively, that were searched before the target was found. A lower value for these metrics indicates a more efficient search. The Overall Search Rate (OSR) is a weighted average of these values, representing the overall search cost:

$$OSR = w_1 \cdot R_r + w_2 \cdot R_c + w_3 \cdot R_i, \tag{8}$$

where w_1, w_2, and w_3 are weights that reflect the relative importance of each search level. A reasonable choice for the weights could be assigned as $w_1 = 0.2$, $w_2 = 0.3$, $w_3 = 0.5$.

We compare our method against two traditional non-semantic search methods: a full Coverage Search and a Random Walk search. The results are shown in Table I.

TABLE I: Search efficiency of different strategies. Lower values indicate higher efficiency (less of the environment was searched).

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

The results in Table I validate the efficiency of the GODHS framework. Guided by both LLMs, our system demonstrates significantly lower search rates across all categories compared to the non-semantic baselines. This indicates that by leveraging hierarchical, semantic guidance, the robot needs to explore a much smaller fraction of the environment to locate the target, confirming that the approach significantly reduces search cost and improves efficiency.

We identified three categories of failure modes: (i) hardware limitations—for example, the manipulator cannot access areas close to the floor; (ii) insufficient common-sense in the LLM—for instance, it may instruct the robot to inspect a non-openable exterior panel of a fridge; and (iii) semantic ambiguity—such as misclassifying a billiard table.

C. Performance Evaluation

In this section, we evaluate the performance of our heuristic-based motion planner, specifically the pose sorting

strategies detailed in Sec. III-B. The experiment is designed to assess the effectiveness of our approach in optimizing the robot's exploration path and reducing execution time in the simulated environment.

To systematically assess the effectiveness of our proposed sorting strategies, we compare four configurations: an unoptimized baseline with no sorting applied to either EE or CH poses; a configuration where only EE poses are optimized via lexicographical sorting; another where only CH poses are optimized using polar angle sorting; and a final setup where both optimizations are applied together.

We evaluate three key performance metrics: the Normalized EE Path Length, which represents the ratio of the total EE travel distance to the theoretical shortest path; the Normalized CH Path Length, defined similarly for the CH poses; and the Execution Time Ratio, which is the ratio of execution time after optimization to that of the unoptimized case, with a lower value indicating better efficiency.

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**

Table II presents the comparative results. The data demonstrates that applying lexicographical sorting to EE poses and polar angle sorting to CH poses independently yield significant improvements in their respective path lengths. 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.

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 multi-modal foundation models [28] and social navigation models [29] to enhance its capabilities.

References

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

[2] 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.

[3] 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.

[4] 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.

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

[6] 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.

[7] L. Zhang, Y. Ling, Z. Bing, F. Wu, S. Haddadin, and A. Knoll, "Tree-based grafting approach for bidirectional motion planning with local subsets optimization," *IEEE Robotics and Automation Letters*, vol. 10, no. 6, pp. 5815–5822, 2025.

[8] K. Cai, R. Laha, Y. Gong, L. Chen, L. Zhang, L. F. Figueredo, and S. Haddadin, "Demonstration to adaptation: A user-guided framework for sequential and real-time planning," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 9871–9878.

[9] L. Zhang, K. Cai, Z. Bing, C. Wang, and A. Knoll, "Genetic informed trees (GIT*): Path planning via reinforced genetic programming heuristics," *Biomimetic Intelligence and Robotics*, vol. 5, no. 3, p. 100237, 2025.

[10] 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.

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

[12] Y. Li, Y. Ma, X. Huo, and X. Wu, "Remote object navigation for service robots using hierarchical knowledge graph in human-centered environments," *Intelligent Service Robotics*, vol. 15, pp. 1–15, 06 2022.

[13] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," 2023.

[14] M. Chang, T. Gervet, M. Khanna, S. Yenamandra, D. Shah, S. Y. Min, K. Shah, C. Paxton, S. Gupta, D. Batra, R. Mottaghi, J. Malik, and D. S. Chaplot, "Goat: Go to any thing," 2023.

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

[16] 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.

[17] 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.

[18] L. Zhang, Z. Bing, Y. Zhang, K. Cai, L. Chen, F. Wu, S. Haddadin, and A. Knoll, "Elliptical k-nearest neighbors - path optimization via coulomb's law and invalid vertices in c-space obstacles," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 12 032–12 039.

[19] Y. Wang, F. Giuliari, R. Berra, A. Castellini, A. D. Bue, A. Farinelli, M. Cristani, and F. Setti, "POMP: pomcp-based online motion planning for active visual search in indoor environments," *CoRR*, vol. abs/2009.08140, 2020.

[20] 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.

[21] 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.

[22] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," 2022.

[23] 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.

[24] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang, S. Gupta, B. P. Majumder, K. Hermann, S. Welleck, A. Yazdanbakhsh, and P. Clark, "Self-refine: Iterative refinement with self-feedback," 2023.

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

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

[27] 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.

[28] R. Bommasani, D. A. Hudson, E. Adeli *et al.*, "On the opportunities and risks of foundation models," 2022.

[29] 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.