

# osmAG-LLM: Zero-Shot Open-Vocabulary Object Navigation via Semantic Maps and Large Language Models Reasoning

Fujing Xie<sup>1</sup>, Sören Schwertfeger<sup>1</sup>, and Hermann Blum<sup>2</sup>

**Abstract**—Recent open-vocabulary robot mapping methods enrich dense geometric maps with pre-trained visual-language features, achieving a high level of detail and guiding robots to find objects specified by open-vocabulary language queries. While the issue of scalability for such approaches has received some attention, another fundamental problem is that high-detail object mapping quickly becomes outdated, as objects get moved around a lot. In this work, we develop a mapping and navigation system for object-goal navigation that, from the ground up, considers the possibilities that a queried object can have moved, or may not be mapped at all. Instead of striving for high-fidelity mapping detail, we consider that the main purpose of a map is to provide environment grounding and context, which we combine with the semantic priors of LLMs to reason about object locations and deploy an active, online approach to navigate to the objects. Through simulated and real-world experiments we find that our approach tends to have higher retrieval success at shorter path lengths for static objects and by far outperforms prior approaches in cases of dynamic or unmapped object queries. We provide our code and dataset at: <https://github.com/xiexiexiaoxiexie/osmAG-LLM>.

**Index Terms**—Semantic Scene Understanding, AI-Enabled Robotics, Vision-Based Navigation

## I. INTRODUCTION

NAVIGATION typically requires mapping and planning at multiple scales. Applications such as last mile delivery, inner-city logistics, or assistive navigation systems require city-level street maps to get to the right building, but then also need to navigate inside buildings to go to a specific room or find a requested object. Furthermore, these tasks need understanding about how our world is structured, which items are usually kept in which locations, the usual layouts

Manuscript received: July, 15, 2025; Revised October, 18, 2025; Accepted December, 2, 2025.

This paper was recommended for publication by Editor Markus Vincze upon evaluation of the Associate Editor and Reviewers’ comments. This work was supported by the Shanghai Frontiers Science Center of Human-centered Artificial Intelligence and German Federal Ministry of Education and Research (BMBF) in the project “Robotics Institute Germany”, grant No. 16ME0999. The work was made possible with the support of a scholarship from the German Academic Exchange Service (DAAD). The experiments of this work were supported by the core facility Platform of Computer Science and Communication, SIST, ShanghaiTech University.

<sup>1</sup>Fujing Xie and Sören Schwertfeger are with the Key Laboratory of Intelligent Perception and Human-Machine Collaboration – ShanghaiTech University, Ministry of Education, China. [xiefj@shanghaitech.edu.cn](mailto:xiefj@shanghaitech.edu.cn)

<sup>2</sup>Hermann Blum is with Robot Perception and Learning Lab, University of Bonn and Lamarr Institute for ML and AI, 53115 Bonn, Germany. [blumh@uni-bonn.de](mailto:blumh@uni-bonn.de)

Digital Object Identifier (DOI): see top of this page.

©2026 IEEE

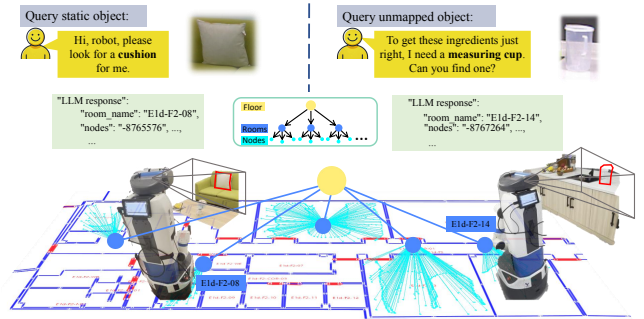


Fig. 1: The semantic-osmAG employed in our method is a hierarchical, topometric map representation enhanced with textual semantic objects (attached to cyan nodes) and room attributes (attached to rooms). By leveraging this map with LLMs, the robot achieves efficient navigation and objects localization—even for objects absent during initial mapping phase (unmapped objects).

and functions of different rooms or spaces, as well as other environment-related information.

Multiple prior works [1]–[3] found that large language models (LLMs) exhibit very good high-level planning capabilities, especially because they already bring knowledge about many semantic priors in our everyday environments. Robots, however, require not only the general knowledge of LLMs, but also grounding to a specific environment. For instance, while a LLM might suggest searching for scissors in a kitchen, providing the robot with a detailed map of a building allows it to find scissors much quicker in the workshop next door.

Previous works on utilizing LLMs for object navigation, such as [4], [5], often rely on language-based instructions like “Bring me the bottom picture that is next to the top of stairs on level one” or “find a bean bag in the office on the first floor” to guide robots. These requests already include a lot of map knowledge in the input to the task, which may not always be known to the user or the environment might be too large and complex for such explicit guidance. In this work we also consider less specified queries, where a robot is tasked to retrieve a certain type of object, even if it might have moved, there are multiple possible ones in the map, or it has not been mapped at all. As shown in Fig. 1, through a compact, scalable map representations that fits into the context windows of LLMs, we can merge semantic priors and map knowledge to successfully cover all these cases.

In this work, we build upon osmAG (Area Graph in Open-

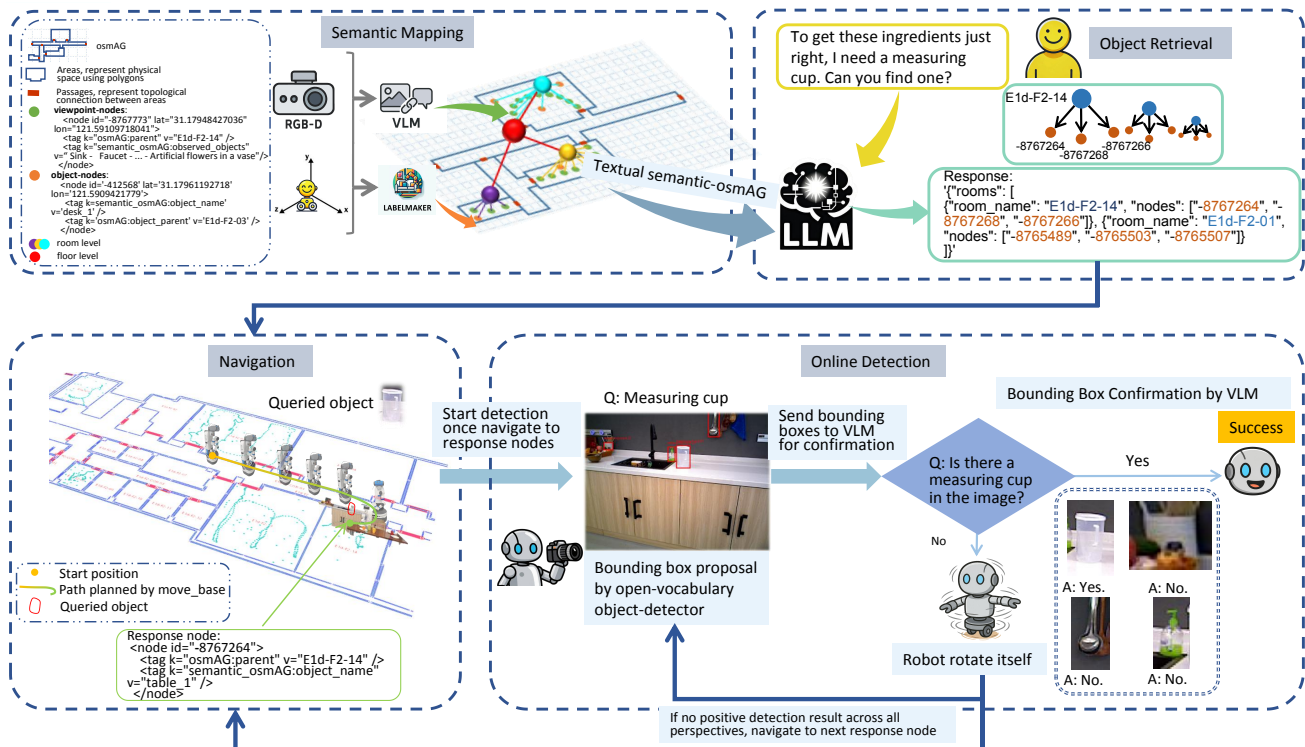


Fig. 2: An overview of our method. We construct a semantic-osmAG offline by augmenting the basic osmAG with two additional keys: object-nodes (extracted via LabelMaker from RGB-D trajectory data) and viewpoint-nodes (processed by a VLM and placed along the trajectory path). When given a human query, the system uses an LLM to generate proposed geometric nodes (response nodes) based on the pre-built textual semantic-osmAG and the query. The robot then navigates to these nodes one by one using ROS move\_base. Once at a response node, an open-vocabulary object detector proposes bounding boxes for the queried object, which are then checked by a VLM to verify if the object is actually present. If the object isn't found, the robot turns to capture additional perspectives. If the object still isn't detected after checking all views at that node, the robot moves to the next response node and repeats the detection process. (Cartoon robot in Online Detection module generated by ChatGPT-4V; prompt: 'robot rotating in place, taking pictures'.)

StreetMap<sup>1</sup> textual format) [6], a framework that adds room-level mapping to OpenStreetMap, and extends it to object-level mapping and navigation. Building on osmAG has two key advantages: (i) our system is directly compatible with city-level navigation [7] and can take advantage of the hierarchies defined in OSM; (ii) OSM is a text-based XML format that can directly be parsed by LLMs, without any necessity to collect large-scale data to train encoders and projectors. In particular, we leverage the LabelMaker pipeline [8] as well as VLMs to populate a map with initial object instances and room descriptions. While we prioritize scalability over completeness of the map, we show that through online detection and decision making, we are able to retrieve unmapped objects with close to the same success rate as mapped objects, highlighting the importance of the context a map provides.

In summary, our contributions are:

- We extend osmAG to object-level semantic mapping, creating a textual semantic map that can be parsed by LLMs.
- We develop an online object retrieval algorithm that combines the planning capabilities and semantic priors of LLMs with online open-vocabulary detection to retrieve

even rare and unusual objects that are not explicitly mapped.

- We conduct extensive experiments in simulation and in a real-world campus building, extending the common test cases with object queries that have been moved, as well as querying for unmapped objects.

To promote reproducibility and further research, we open-source our code, dataset, and related resources.

## II. RELATED WORKS

### A. Semantic Maps for LLM-Guided Robotics

Prior works in semantic maps for robotics fall into two categories: explicit and implicit representations. Explicit maps use predefined semantic labels (e.g., object classes), offering interpretability but limited flexibility. For example, [9] construct a 2D semantic map from projecting Mask-RCNN-based instance segmentations. Tag Map [10] represent objects as [id] - [tag] pairs, enabling reasoning with LLMs but lacking geometric structure. Guide-LLM [11] employ a text-based topological map for route planning, assisting visually impaired users through LLM-guided navigation. Implicit maps leverage vision-language models (e.g., CLIP [12]) for open-vocabulary scene understanding. For example, OpenScene [13] and OpenMask3D [14] embed CLIP-aligned features

<sup>1</sup><https://www.openstreetmap.org/>

**IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.**

into 3D point clouds or segmented instances, ConceptGraphs [15] and HOV-SG [5] further organize instances into 3D scene graphs, with the latter introducing hierarchical open-vocabulary relationships for language-grounded navigation. MoMa-LLM [16] utilize a semantic scene graph for long-horizon mobile manipulation, where the robot updates its semantic representation online and uses LLM for task planning. Unlike these approaches, our method uses semantic-osmAG, a text-based explicit map that contains rich semantic information while retaining hierarchical and topometric structure, enabling traditional robotic localization [17] and navigation [18].

### B. Object Navigation Without Prior Maps

Prior research in object-goal navigation has explored methods that operate without prior environmental maps. Early approaches, such as SemExp [19], relied on projective geometry, using Mask R-CNN to extract semantic features from RGB observations and projecting them into 3D voxel maps for reinforcement learning (RL)-based exploration. Subsequent works address generalization limitations by incorporating vision-language models. For instance, CoWs [20] implement nearest-frontier exploration guided by CLIP’s zero-shot recognition. Similarly, VLFM [21] introduce open-vocabulary frontier scoring, embedding observations via BLIP-2 [22] and computing semantic similarity to target objects to prioritize exploration paths. GAMap [23] further enhance this approach by integrating object affordances (e.g., “graspable surfaces”) into incrementally constructed maps to guide exploration.

There are also works focused on improving spatial reasoning using LLMs. For example, ESC [24] employ LLMs as rule-based frontier selectors, leveraging commonsense reasoning to infer spatial relationships between target objects and common objects or room layouts. Likewise, [25] utilize LLMs’ commonsense knowledge to predict object locations.

However, these methods do not rely on pre-built maps, which can lead to challenges in scenarios such as locating a toilet behind a closed bathroom door or navigating long corridors where doorways are not immediately visible. In contrast, our method leverages pre-built maps to address these limitations.

### C. Object Navigation with Prior Maps

Prior research has also explored object-goal navigation using pre-built maps. ConceptGraphs [15] generate 3D scene graphs by projecting class-agnostic segments into metric maps, embedding them with CLIP, and inferring spatial relationships via LLMs. Other graph-based methods like SEEK [26] utilize dynamic scene graphs (DSGs) derived from environmental blueprints, training a neural network to estimate the probability of finding the target object across spatial elements in the graph. HOV-SG [5] construct a hierarchical open-vocabulary 3D scene graph that enables hierarchical query and object navigation. Unlike prior methods that rely on dense representations, our approach: 1. avoids detailed 3D environment mapping to ensure long-term usability, 2. uses powerful VLMs to summarize environmental objects, 3. maintains a lightweight textual map with hierarchy and topology that preserves key semantic relationships for LLM-based reasoning.

## III. APPROACH

This work tackles the challenge of map-guided object navigation by combining semantic maps (semantic-osmAG, based on osmAG [6] as shown in Fig. 1) with large language models. As illustrated in Fig. 2, our approach begins by augmenting the basic osmAG map using RGB-D trajectories to build a semantic-rich map representation for subsequent tasks (Section III-A). Next, given a human query, an LLM is leveraged to infer probable map nodes corresponding to the target object (Section III-B). Finally, the robot navigates toward the nodes (Section III-C), with navigation further refined through real-time online object detection (Section III-D).

### A. Semantic Environment Mapping

1) *Map Representation*: While traditional maps (e.g., occupancy grids, point clouds) excel at spatial representation, their raw numeric form offers little semantic structure and is therefore difficult for a large language model to understand. We instead employ semantic-osmAG—an indoor, semantic extension of the osmAG map—because it provides four decisive advantages:

- 1) *Human-LLM Interpretable Format*: osmAG uses OSM standards, with editable key-value properties for nodes and rooms, naturally readable by both humans and LLMs without extra conversion.
  - 2) *Compact Representation*: As Section III-A2 describes, osmAG encodes areas (rooms) as simple polygons, objects as single nodes with pure text format, enabling large environment deployment and sharing through OSM server.
  - 3) *Hierarchical Object Retrieval*: The osmAG map uses parent tags to represent which objects are contained in which rooms, enabling LLMs to reason about probable locations. For example, when searching for a hot air gun, the LLM recognizes it more likely belongs in a robotics-equipped lab than a kitchen, and directs the search accordingly. Once the target room is selected, the same hierarchy is used to identify specific nodes within the room. This closely mimics human search behavior by allowing the system to narrow the search from a broad room-level down to precise locations, rather than relying on memorizing the entire environment’s point cloud.
  - 4) *Minimal Update Required*: Through semantic filtering, we can exclusively utilize permanent infrastructure (e.g., walls, doors) for osmAG based robot localization [17] and navigation [18] and avoid saving pixel level object representations. Therefore, semantic-osmAG naturally adapts to dynamic environments and requires minimal updates.
- 2) *osmAG*: osmAG[6] is a hierarchical, topometric semantic map representation based on the Area Graph concept [27]. As illustrated in Fig. 1 and used in our experiments (Fig. 4), osmAG models physical spaces (e.g., rooms, corridors) as polygon-based areas and uses passages (e.g., doors) to represent connections between areas. Moreover, osmAG employs a ‘parent’ tag to explicitly represent hierarchical relationships between spaces (e.g., rooms within a building



Fig. 3: Figure (a) shows a pre-built navigation graph from [5] for reference, while (b)-(d) show our navigation process. With query ‘couch in the living room’: (b) initial setup with an osmAG-rendered occupancy map (walls/doors only), where red rectangles mark ground truth and yellow rectangles show response nodes (green circles highlighting their sequence). (c)-(d) The robot progressively perceives the environment, navigating nodes (1→3) and replanning upon collisions. Our navigation strategy produces more direct paths compared to pre-built navigation graph in (a), and through progressive environmental perception, achieves long-term operation with minimal map updates required.

or floor). osmAG can be constructed from various sources, including 3D point clouds [28], occupancy grid maps [27], or CAD files [6], [29]. Prior works have established osmAG’s utility for core robotic tasks: robust LiDAR-based localization [17] and intelligent navigation [18]. While osmAG originally lacked the granularity to locate specific objects (e.g., a hot air gun), Section III-A3 presents our semantic enrichment method to address this limitation.

3) *Semantic Enrichment*: To integrate semantic information into osmAG, we developed a pipeline for generating semantic nodes from RGB-D data. We generate the object-nodes by first utilizing LabelMaker[8] to process RGB-D trajectories through ensemble voting across state-of-the-art models to produce point clouds for each object category. Next, we use Mask3D’s [30] instance masks to isolate individual instances from category point clouds. We calculate each instance’s centroid and embed it as a ‘node’ (with longitude/latitude coordinates) within the osmAG structure with semantic\_osmAG:object\_name as tag’s key as illustrated in Fig. 2. However, LabelMaker inherits the categorical limitations of its underlying models, failing to recognize objects outside standard datasets like ScanNet200 [31], which limits its ability to process unrestricted natural language queries about objects (e.g., robot dog). To address this, we augment the system with ChatGPT’s [32] visual query understanding capability. By prompting with ‘What objects are in this image?’ at capture locations, we generate viewpoint-nodes at the image’s captured location with tag key as semantic\_osmAG:observed\_object and value as the response of VLM as illustrated in Fig. 2. This hybrid approach – combining geometric instance extraction and open-vocabulary visual language response – generates part of the nodes with exact geometric location and gives part of the nodes open-vocabulary capability from a more intelligent model, overcoming categorical rigidity inherent in LabelMaker alone.

Beyond embedding geometric object nodes, we generate semantic room descriptions by prompting a VLM with ‘Describe the image.’ for all images captured in a room. The resulting descriptions are then summarized by an LLM and attached directly to the corresponding area elements in the semantic-

osmAG map.

### B. Object Retrieval

When processing a human query, our method generates a structured LLM prompt that: (1) explains semantic-osmAG’s representation, (2) defines the object retrieval task objectives, and (3) provides a simplified semantic-osmAG containing only semantic nodes and hierarchical relationships (excluding coordinates and polygons).

To optimize search efficiency and align with practical navigation patterns, we specify to the LLM to generate results using a JSON format where response nodes (which can be either viewpoint or object nodes) are organized by room (max 3 nodes per room and max 3 rooms total), ordered by decreasing predicted likelihood as shown in Fig. 2. This room-centric approach mimics human search behavior - checking one room completely before moving to the next, rather than jumping between individual nodes based solely on probability.

### C. Navigation

As illustrated in Fig. 3, unlike methods like HOV-SG [5] using pre-built navigation graphs, we dynamically navigate via online perception from rendered osmAG. We first convert osmAG to an occupancy grid (area polygons as occupied, passages as free). The system then sequentially visits response nodes from LLM: (1) setting navigation goals from sorted nodes, (2) planning paths via A\* on the occupancy grid, and (3) continuously updating the rendered occupancy grid map using real-time depth data and triggering path replanning when collisions are detected in the updated map. By building the grid only from permanent structures and adapting online, we avoid needing map updates when environments change.

### D. On-line Object Detection

To ensure robust adaptation to environmental changes and verify object presence at response nodes, our system performs real-time detection upon arrival. As shown in Fig. 2, when the robot arrives at a response node, it captures images and executes a two-stage verification process. We run open-vocabulary object-detectors (Yolo-world [33] or DINO-X [34]) on the captured images to generate a list of bounding boxes ranked by confidence. Each box is cropped and validated by a VLM (StepFun [35]) using the prompt ‘‘Is there a [object] here?’’, with the highest-confidence positive detection returned as the final result. We find that this two-step approach is more reliable with long-tail object queries where the detectors would otherwise return too many false positives. If no object is detected, the robot rotates its body or head to capture additional viewpoints, repeating the verification process until either (1) the target object is successfully identified, or (2) all potential viewpoints at the current node have been exhausted without positive confirmation, at which point it proceeds to the next response node.

## IV. EXPERIMENTS

We evaluated our approach on eight scenes from the HM3D-SEM dataset [36] (consistent with our baseline) as well as on a real-world dataset collected in our campus building. In

## IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE I: Object Retrieval &amp; Goal Navigation from Language Queries on HM3DSEM

Query Type	Method	R-RSR	O-RSR <sub>top5</sub>			O-RSR <sub>top1</sub>			AMD [m]	DIR	APL [m]
			@1m	@2m	@3m	@1m	@2m	@3m			
(o,r,f)	HOV-SG	0.27	<b>0.31</b>	0.39	0.44	<b>0.20</b>	0.28	0.33	8.85	0.05	20.15
	our method	<b>0.66</b>	0.25	<b>0.59</b>	<b>0.70</b>	0.16	<b>0.37</b>	<b>0.54</b>	<b>4.65</b>	<b>0.27</b>	<b>12.32</b>
(o,r)	HOV-SG	0.32	<b>0.35</b>	0.44	0.49	<b>0.20</b>	0.28	0.33	7.87	0.06	16.81
	our method	<b>0.67</b>	0.28	<b>0.58</b>	<b>0.68</b>	0.18	<b>0.37</b>	<b>0.55</b>	<b>5.27</b>	<b>0.25</b>	<b>11.54</b>
(o)	HOV-SG	0.65	<b>0.76</b>	0.82	0.88	<b>0.52</b>	<b>0.64</b>	<b>0.70</b>	<b>3.10</b>	0.05	22.44
	our method	<b>0.83</b>	0.47	<b>0.83</b>	<b>0.90</b>	0.28	0.50	0.69	3.37	<b>0.31</b>	<b>11.36</b>

Section IV-A, we justify our selection of HOV-SG[5] and ConceptGraph[15] as baselines. Section IV-B details our evaluation metrics, followed by Habitat simulation experiments using the HM3D-SEM dataset in Section IV-C, and real-world experiments in Section IV-D.

### A. Baseline

Current object-aware mapping approaches suffer from key limitations: Hydra [37] and SEEK [26] lack open-vocabulary support, restricting their utility for real-world queries (e.g., “hot air gun”). VLMaps [38] fails to incorporate hierarchical representations, potentially compromising scalability in large environments.

We select HOV-SG as our baseline because it shares fundamental design principles with our approach. Both methods: (1) rely on pre-built semantic maps for environment representation, (2) utilize natural language object queries as their primary input without describing how to get to the location, (3) have the same input mapping modalities (RGB-Depth-Pose data pairs), and (4) employ a hierarchical map structure.

To ensure a fair comparison, we directly incorporate HOV-SG’s query prompt formulation in our experiments using the HM3D-SEM dataset. In addition to HOV-SG, we include ConceptGraph [15] as an additional baseline in our real-world experiments to provide a more comprehensive evaluation. On the HM3D-SEM dataset, we refer to the evaluation in [5] from which we identify HOV-SG as the stronger baseline for our experiments.

We note that the original HOV-SG’s output point clouds are not room-aware, causing the agent to frequently jump between rooms (e.g., leaving Room 1 and later returning to it), thereby unnecessarily increasing the path length. To mitigate this, we improve the baseline by re-sorting the top 5 HOV-SG predictions to prioritize visiting all nodes in one room before moving to another.

### B. Metrics

We employ the following metrics to evaluate performance across both simulated and real-world environments:

- **Room Retrieval Success Rate (R-RSR)**: Percentage of queries where the top prediction’s room contains at least one ground truth instance of the target object.
- **Object Retrieval Success Rate (O-RSR<sub>top-n</sub>@k)**: Percentage of queries where at least one top- $n$  prediction ( $n \in \{1, 5\}$ ) is within  $k$  meters (Euclidean distance) of ground truth. We test  $k \in \{1, 2, 3\}$  to accommodate positional variance between our 2D image-derived nodes and HOV-SG’s 3D point clouds.

- **Average Minimum Distance (AMD)**: Computes the mean distance error of the closest prediction among the top-5 results to evaluate object localization precision.
- **Average Path Length (APL)**: Mean navigation path length that the robot drove (in meters) for queries where both our method and baseline successfully retrieve the target (within 1m). Measures navigation efficiency.
- **Detection Improvement Rate (DIR)**: Percentage of queries where initially failed retrievals (where all top-5 predicted nodes exceed 1m from ground truth) that are successfully recovered through online detection at response nodes.
- **Map size**: Compares the representation size of our method and HOV-SG.

### C. HM3D-SEM Datasets Experiment

1) *Setup*: We evaluate our approach on 8 diverse HM3D-SEM scenes (00824, 00829, 00843, 00861, 00862, 00873, 00877, 00890) which span multiple rooms and floors. Each scene is tested with 5 randomly chosen starting positions, with the Habitat-simulated robot equipped with an RGB-D camera for perception. The base osmAG map is created manually using JOSM (Java OpenStreetMap Editor)<sup>2</sup> tool. We then enhance it semantically using LabelMaker and ChatGPT-4V, processing the same RGB-D trajectories as HOV-SG to generate the final representation shown in Fig. 3 (b). Navigation follows our defined approach (Section III-C, Fig. 3). Online detection is implemented through a two-stage process: YOLO-World generates initial bounding box proposals, followed by StepFun’s confirmation reasoning. We generate queries using the same approach as HOV-SG with three granularity levels for 20 common objects: 1. object (o): “pillow”; 2. object + room (o,r): “pillow in the living room”; 3. object + room + floor (o,r,f): “pillow in the living room on floor 0”.

2) *Results*: Table I presents comprehensive evaluation results across three query granularities, our method demonstrates significant improvements in key metrics over the HOV-SG baseline:

- Superior room and object retrieval: Notably, we achieve 144% higher Room Retrieval Success Rate (R-RSR) for complex (o,r,f) queries (0.66 vs. 0.27) and maintain better performance at practical operating distances (O-RSR@2m/3m), despite HOV-SG’s advantage in precise close-range (@1m) object retrieval due to its direct point-cloud outputs. The AMD metric confirms our predicted nodes are closer to ground truth positions in (o,r) and (o,r,f) query types.

<sup>2</sup><https://josm.openstreetmap.de/>

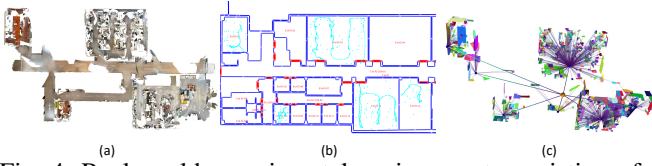


Fig. 4: Real-world experimental environment consisting of: a conference room (upper left), a student office (upper right), a professor’s office (lower left), a robotics lab (lower middle), and a relaxation lounge (lower right). (a) Data collected using an Apple scanner with the “3D Scanner App”; (b) the semantic-osmAG map; (c) the HOV-SG generated from the collected data.

- Hierarchical advantage: Both methods exhibit performance degradation as query complexity increases, but our approach demonstrates significantly more graceful degradation compared to HOV-SG.
- Effective online detection compensation: The online detection system effectively compensates for retrieval imperfections, as evidenced by substantially higher DIR (Detection-Improvement Rate).
- Efficient navigation: The Average Path Length (APL) metric shows that our method reduces traversal distance by 31-49% across all conditions. This confirms that navigation using osmAG is more effective than navigation using the pre-build navigation graph of the baseline (illustrated in Fig. 3).
- Significantly smaller storage requirements: As shown in Table II, our method requires substantially less storage space than the baseline (while HOV-SG’s representation scales with computation time rather than context size, this map-size comparison highlights the compactness and storage efficiency of our method).

TABLE II: Representation Size Comparison (MB)

	HM3D-SEM (8 scenes)	Real Dataset
HOV-SG	1493	129
Our method	3.2	0.62

#### D. Real-World Environment Experiment

1) *Setup*: We conducted experiments in a large and cluttered campus building spanning multiple functional areas: a conference room (46  $m^2$ ), a robotics lab (120  $m^2$ ), a student office (147  $m^2$ ), a professor’s office (15  $m^2$ ), and a relaxation lounge (82  $m^2$ ) (Fig. 4). To scale our approach to larger environments, we further propose a two-stage query variant. The first stage uses only the room-level descriptions for coarse room selection, while the second stage provides the detailed map of the chosen room to select specific target nodes. To analyze the trade-off in map complexity, we also investigated the impact of viewpoint node density—examining how performance varies when too few nodes may provide insufficient context versus when too many may confuse the LLM.

For scanning, we exclusively used an Apple scanner with the ‘3D Scanner App’<sup>3</sup>. Unlike the high-fidelity HM3D-SEM dataset, our environment was intentionally mapped with

<sup>3</sup><https://3dscannerapp.com/>



Fig. 5: Experimental objects used for real-world evaluation: The first row shows static objects, the second row contains relocated objects, and the final row displays unmapped objects absent from the map. Successful detections are shown with automatically generated bounding boxes during experiments, while text overlays indicate object names and instance counts in our environment (blue = success, red = failure).

lightweight scanning to evaluate practical deployment feasibility, resulting in less detailed spatial representations (Fig. 4a). The robotic platform consisted of a Fetch mobile manipulator equipped with: A base-mounted 2D LiDAR for localization and navigation (using ROS AMCL and move\_base packages), a head pan link capable of 180-degree rotation to capture diverse viewpoints, and a head-mounted RealSense RGB-D camera for visual perception. As the student office contains steps that are not traversable with Fetch, we implemented two methodological adaptations: 1) Manual camera positioning at designated response nodes for object detection. 2) Use of planned (rather than executed) global paths from move\_base for Average Path Length (APL) metrics. These adaptations were applied to both our method and HOV-SG, maintaining comparative fairness. For both methods, we further simplify testing by skipping physical navigation when the query object was definitively absent from a room, again using move\_base’s global path as Path Length.

2) *Queries*: To evaluate both methods under realistic conditions, we conducted 30 experiments divided into three distinct categories (shown in Fig. 5): Static Objects (SO, e.g., sink/TV) remaining approximately at mapped positions; Relocated Objects (RO, e.g., screwdriver/robot dog) moved from initial locations; and Unmapped Objects (UO, e.g., presentation remote/measuring cup) absent during mapping. Each category (SO, RO, UO) contained six unique single-location objects and four multi-instance objects, totaling 11, 12, and 13 objects per category respectively.

3) *Results*: Our real-world experiments demonstrate consistent advantages over baseline (Table III):

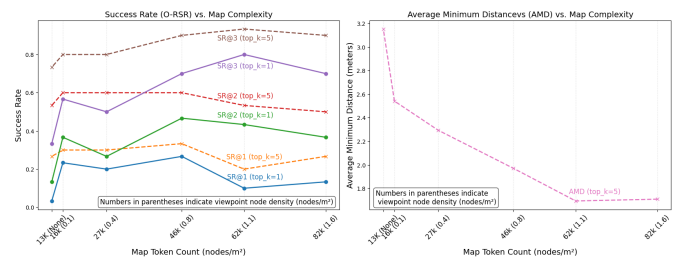


Fig. 6: Object retrieval performance versus map complexity, represented by map token count and its corresponding viewpoint node density (parenthetical). (Left) Object Retrieval Success Rate (O-RSR) and (Right) Average Minimum Distance (AMD).

## IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE III: Object Retrieval &amp; Goal Navigation from Language Queries on Real Data

Object Type	Method	R-RSR	O-RSR <sub>top5</sub>			O-RSR <sub>top1</sub>			AMD [m]	DIR	APL [m]
			@1m	@2m	@3m	@1m	@2m	@3m			
static (SO)	ConceptGraph	0.50	<b>0.50</b>	0.50	0.60	<b>0.20</b>	0.30	0.40	3.52 (3.52)	/	/
	HOV-SG	0.40	0.20	<b>0.70</b>	<b>0.80</b>	0.00	0.30	0.30	1.84 (1.84)	0.40 (0.40)	47.00
	our method	<b>1.00</b>	0.30	0.50	<b>0.80</b>	<b>0.20</b>	<b>0.40</b>	<b>0.80</b>	<b>1.78 (1.78)</b>	0.40 (0.40)	<b>18.13</b>
relocated (RO)	ConceptGraph	0.50	0.20	0.30	0.50	0.00	0.10	0.20	5.51 (5.86)	/	/
	HOV-SG	0.40	<b>0.30</b>	0.50	0.50	0.00	0.20	0.20	5.36 (5.79)	0.00 (0.00)	22.05
	our method	<b>1.00</b>	<b>0.30</b>	<b>0.70</b>	<b>0.90</b>	<b>0.20</b>	<b>0.60</b>	<b>0.80</b>	<b>1.63 (1.72)</b>	<b>0.70 (0.78)</b>	<b>20.43</b>
unmapped (UO)	ConceptGraph	0.40	<b>0.10</b>	0.20	0.30	0.00	<b>0.10</b>	0.20	8.35 (8.83)	/	/
	HOV-SG	0.20	0.00	<b>0.30</b>	0.50	0.00	0.00	0.20	7.42 (7.63)	0.30 (0.33)	52.03
	our method	<b>1.00</b>	0.00	<b>0.30</b>	<b>0.80</b>	0.00	<b>0.10</b>	<b>0.60</b>	<b>2.25 (2.33)</b>	<b>0.80 (0.78)</b>	<b>30.68</b>

Values in parentheses indicate: results after excluding inaccessible rooms (Section IV-D1), maintaining the same trends as overall results.

- Superior room and object retrieval: Our method achieves 100% room retrieval, surpasses baselines in O-RSR metrics as distance increases, and maintains lower AMD (nodes closer to ground truth).
- Effective online detection compensation: Our higher DIR provides strong evidence that our online detection approach effectively compensates for sparse mapping requirements while eliminating the need for precise 3D point cloud data saved in the map.
- Efficient navigation: The Average Path Length (APL) metric shows our method reduces path length across all conditions.
- Environmental robustness: Furthermore, our method significantly outperforms baselines on relocated and unmapped objects, highlighting its robustness to environmental variability.
- Two-stage query performance: The results in Table IV reveal a surprising effectiveness of this approach for static objects. For relocated or unmapped objects, the two-stage approach is less successful than when providing context of all the room immediately, but still on par or better than the baselines. Overall these indicate promising scalability of osmAG-LLM for very large environments.
- Viewpoint node density: Our experiment reveals a non-linear relationship between viewpoint node density and performance (Fig. 6). A small number of nodes yields substantial performance gains, offering high reward. However, beyond a certain threshold, additional nodes provide diminishing returns and may even slightly degrade performance due to potential information overload. Overall the density does not seem to require critical tuning as long as some viewpoint nodes for every room are provided.

4) *Ablation Studies*: To evaluate the contributions of the two key components in our map–object nodes and viewpoint nodes—we conduct an ablation study using real-world data (Table IV). First, we remove all object nodes generated by LabelMaker. This degrades RSR and AMD performance, since the system loses precise object location information. Next, we eliminate all viewpoint nodes generated by the VLM, which leads to rapid performance degradation, since our language queries contain many objects outside our detection model’s fixed category set. Our results demonstrate that both node types are essential, as removing either component leads to substantially worse performance compared to the complete method.

TABLE IV: Comparison of Method Variants on Real Data

Object Type	Method	R-RSR	O-RSR <sub>top5</sub>			O-RSR <sub>top1</sub>			AMD [m]
			@1m	@2m	@3m	@1m	@2m	@3m	
SO	Ours	<b>1.0</b>	0.3	0.5	0.8	0.2	0.4	<b>0.8</b>	1.78
	Ours-2stages	<b>1.0</b>	<b>0.4</b>	<b>0.8</b>	<b>1.0</b>	<b>0.3</b>	<b>0.5</b>	<b>0.8</b>	<b>1.30</b>
	w/o ON	<b>1.0</b>	0.3	0.5	0.7	<b>0.3</b>	0.4	0.6	2.14
RO	Ours	<b>1.0</b>	<b>0.3</b>	<b>0.7</b>	<b>0.9</b>	0.2	<b>0.6</b>	<b>0.8</b>	<b>1.63</b>
	Ours-2stages	0.7	<b>0.3</b>	0.5	0.7	0.2	0.4	0.6	7.44
	w/o ON	0.9	<b>0.3</b>	<b>0.7</b>	0.8	<b>0.3</b>	0.5	0.6	2.00
UO	Ours	<b>1.0</b>	0.0	<b>0.3</b>	<b>0.8</b>	0.0	<b>0.1</b>	<b>0.6</b>	<b>2.25</b>
	Ours-2stages	0.9	0.0	0.2	0.7	0.0	<b>0.1</b>	<b>0.6</b>	4.37
	w/o ON	0.9	0.0	0.1	0.4	0.0	<b>0.1</b>	0.2	4.73
	w/o VN	0.7	0.0	0.2	0.5	0.0	0.0	0.1	6.24

“w/o ON” = without object nodes, “w/o VN” = without viewpoint nodes.

5) *Failure Analysis*: Our experiments resulted in 9 failed cases out of 30 trials, as shown in Fig. 7. In six cases (displayed in the first two columns of Fig. 7), the robot reached positions where the target objects were visible but still failed to retrieve them. The remaining three failures happened when the robot stopped at nodes where the queried objects weren’t visible, yet the detection method incorrectly reported positive identifications. Therefore, the system achieved 90% correct robot positioning (27/30), with detection errors accounting for most failures.

## V. CONCLUSION AND FUTURE WORK

In this letter, we propose the osmAG-LLM framework that combines a lightweight, text-based semantic map with LLM reasoning and active object search to perform object-goal navigation. We show that this approach outperforms established,

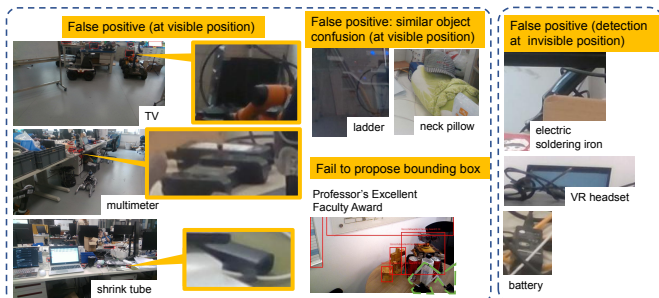


Fig. 7: Our real-world experiments encountered nine failure cases, split by whether the robot could see the objects at those nodes. Six occurred at positions where the queried object was visible, including three detection errors, two confusion cases (misidentifying a ladder reflection as a ladder and a hugging pillow as a neck pillow), and one failure to generate bounding box for a transparent object. The other three failures resulted from false positive detections at non-visible nodes.

## IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

more high-fidelity mapping approaches and specifically shines when retrieving relocated or unmapped objects. While these two cases are already more realistic, steps for our future research include searching for objects in concealed spaces such as drawers, as well as adding the ability to grasp the objects for full retrieval.

## REFERENCES

- [1] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," in *International Conference on Learning Representations (ICLR)*, 2023.
- [2] G. Wang, Y. Xie, Y. Jiang, A. Mandelkar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar, "Voyager: An open-ended embodied agent with large language models," *arXiv preprint arXiv:2305.16291*, 2023.
- [3] J. S. Park, J. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, "Generative agents: Interactive simulacra of human behavior," in *Proceedings of the 36th annual acm symposium on user interface software and technology*, 2023, pp. 1–22.
- [4] Y. Qi, Q. Wu, P. Anderson, X. Wang, W. Y. Wang, C. Shen, and A. v. d. Hengel, "Reverie: Remote embodied visual referring expression in real indoor environments," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9982–9991.
- [5] A. Werby, C. Huang, M. Büchner, A. Valada, and W. Burgard, "Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation," in *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*, 2024.
- [6] D. Feng, C. Li, Y. Zhang, C. Yu, and S. Schwertfeger, "Osmag: Hierarchical semantic topometric area graph maps in the osm format for mobile robotics," in *2023 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2023, pp. 1–7.
- [7] M. Hentschel and B. Wagner, "Autonomous robot navigation based on openstreetmap geodata," in *13th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2010, pp. 1645–1650.
- [8] S. Weder, H. Blum, F. Engelmann, and M. Pollefeys, "Labelmaker: automatic semantic label generation from rgb-d trajectories," in *2024 International Conference on 3D Vision (3DV)*. IEEE, 2024, pp. 334–343.
- [9] T. Gervet, S. Chintala, D. Batra, J. Malik, and D. S. Chaplot, "Navigating to objects in the real world," *Science Robotics*, vol. 8, no. 79, p. eadf6991, 2023.
- [10] M. Zhang, K. Qu, V. Patil, C. Cadena, and M. Hutter, "Tag map: A text-based map for spatial reasoning and navigation with large language models," *arXiv preprint arXiv:2409.15451*, 2024.
- [11] S. Song, S. Kodagoda, A. Gunatilake, M. G. Carmichael, K. Thiagarajan, and J. Martin, "Guide-llm: An embodied llm agent and text-based topological map for robotic guidance of people with visual impairments," *arXiv preprint arXiv:2410.20666*, 2024.
- [12] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PmLR, 2021, pp. 8748–8763.
- [13] S. Peng, K. Genova, C. Jiang, A. Tagliasacchi, M. Pollefeys, T. Funkhouser *et al.*, "Openscene: 3d scene understanding with open vocabularies," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 815–824.
- [14] A. Takmaz, E. Fedele, R. W. Sumner, M. Pollefeys, F. Tombari, and F. Engelmann, "Openmask3d: Open-vocabulary 3d instance segmentation," *arXiv preprint arXiv:2306.13631*, 2023.
- [15] Q. Gu, A. Kuwajerwala, S. Morin, K. M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa *et al.*, "Concept-graphs: Open-vocabulary 3d scene graphs for perception and planning," *arXiv preprint arXiv:2309.16650*, 2023.
- [16] 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*, 2024.
- [17] F. Xie and S. Schwertfeger, "Robust lifelong indoor lidar localization using the area graph," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 531–538, 2023.
- [18] F. Xie, J. Zhang, and S. Schwertfeger, "Intelligent lidar navigation: Leveraging external information and semantic maps with llm as copilot," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2025.
- [19] D. S. Chaplot, D. P. Gandhi, A. Gupta, and R. R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," *Advances in Neural Information Processing Systems*, vol. 33, pp. 4247–4258, 2020.
- [20] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song, "Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 23 171–23 181.
- [21] N. Yokoyama, S. Ha, D. Batra, J. Wang, and B. Bucher, "Vlfm: Vision-language frontier maps for zero-shot semantic navigation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 42–48.
- [22] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International conference on machine learning*. PMLR, 2023, pp. 19 730–19 742.
- [23] H. Huang, Y. Hao, C. Wen, A. Tzes, Y. Fang *et al.*, "Gamap: Zero-shot object goal navigation with multi-scale geometric-affordance guidance," *Advances in Neural Information Processing Systems*, vol. 37, pp. 39 386–39 408, 2024.
- [24] K. Zhou, K. Zheng, C. Pryor, Y. Shen, H. Jin, L. Getoor, and X. E. Wang, "Esc: Exploration with soft commonsense constraints for zero-shot object navigation," in *International Conference on Machine Learning*. PMLR, 2023, pp. 42 829–42 842.
- [25] P. Arjun, A. Melnik, and G. C. Nandi, "Cognitive planning for object goal navigation using generative ai models," in *NeurIPS 2024 Workshop on Open-World Agents*, 2024.
- [26] M. F. Ginting, S.-K. Kim, D. D. Fan, M. Palieri, M. J. Kochenderfer, and A. Akbar Agha-mohammadi, "SEEK: Semantic reasoning for object goal navigation in real world inspection tasks," in *Proc. of Robotics: Science and Systems*, 2024.
- [27] J. Hou, Y. Yuan, and S. Schwertfeger, "Area graph: Generation of topological maps using the voronoi diagram," in *2019 19th International Conference on Advanced Robotics (ICAR)*. IEEE, 2019, pp. 509–515.
- [28] Z. He, H. Sun, J. Hou, Y. Ha, and S. Schwertfeger, "Hierarchical topometric representation of 3d robotic maps," *Autonomous Robots*, vol. 45, no. 5, pp. 755–771, 2021.
- [29] J. Zhang, S. Wu, X. Ma, and S. Schwertfeger, "Generation of indoor open street maps for robot navigation from cad files," *arXiv preprint arXiv:2507.00552*, 2025.
- [30] J. Schult, F. Engelmann, A. Hermans, O. Litany, S. Tang, and B. Leibe, "Mask3d: Mask transformer for 3d semantic instance segmentation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 8216–8223.
- [31] D. Rozenberszki, O. Litany, and A. Dai, "Language-grounded indoor 3d semantic segmentation in the wild," in *European Conference on Computer Vision*. Springer, 2022, pp. 125–141.
- [32] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.
- [33] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, "Yolo-world: Real-time open-vocabulary object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 16 901–16 911.
- [34] T. Ren, Y. Chen, Q. Jiang, Z. Zeng, Y. Xiong, W. Liu, Z. Ma, J. Shen, Y. Gao, X. Jiang *et al.*, "Dino-x: A unified vision model for open-world object detection and understanding," *arXiv preprint arXiv:2411.14347*, 2024.
- [35] StepFun, "step-1o-turbo-vision," StepFun API documentation, <https://platform.stepfun.com/docs/llm/vision>, 2025, accessed 2025-06-13; visual understanding model for text, image, and video input.
- [36] K. Yadav, R. Ramrakhya, S. K. Ramakrishnan, T. Gervet, J. Turner, A. Gokaslan, N. Maestre, A. X. Chang, D. Batra, M. Savva *et al.*, "Habitat-matterport 3d semantics dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4927–4936.
- [37] N. Hughes, Y. Chang, and L. Carlone, "Hydra: A real-time spatial perception system for 3d scene graph construction and optimization," *arXiv preprint arXiv:2201.13360*, 2022.
- [38] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 10 608–10 615.