Article | Received 19 May 2025; Revised 16 October 2025; Accepted 6 November 2025; Published 26 November 2025
https://doi.org/10.55092/rl20250011

Vision language models can parse floor plan maps

David DeFazio†, Hrudayangam Mehta†, Meng Wang†, Ping Yang, Jeremy Blackburn and Shiqi Zhang*

School of Computing, Binghamton University, Binghamton, New York, USA

† These authors contributed equally to this work.

* Correspondence author; E-mail: zhangs@binghamton.edu.

Highlights:

- This article shows that current vision-language models (VLMs) can be used for parsing floor plan maps and planning for indoor navigation tasks.
- The proposed system was evaluated using real floor plan maps and two multimodal foundation models, including GPT-4o and Claude-3.5 Sonnet.
- Results demonstrated the effectiveness of VLM-based robot navigation systems, highlighted practical suggestions, and illustrated applications on a quadruped robot.

Abstract: Vision language models (VLMs) can simultaneously reason about images and texts to tackle many tasks, from visual question answering to image captioning. This paper focuses on map parsing, a novel task that is unexplored within the VLM context and particularly useful to mobile robots. Map parsing requires understanding not only the labels but also the geometric configurations of a map, *i.e.*, what areas are like and how they are connected. To evaluate the performance of VLMs on map parsing, we prompt VLMs with floor plan maps to generate task plans for complex indoor navigation. Our results demonstrate the remarkable capability of VLMs in map parsing, with a success rate of 0.96 in tasks requiring a sequence of nine navigation actions, e.g., approaching and going through doors. Other than intuitive observations, e.g., VLMs do better in smaller maps and simpler navigation tasks, there was a very interesting observation that its performance drops in large open areas. We provide practical suggestions to address such challenges as validated by our experimental results. Webpage: https://sites.google.com/view/vlm-floorplan/.

Keywords: task planning; mobile robots; vision language models; robot navigation

1. Introduction

A key to mobile robotics is a deep understanding of the geometric configuration of the world that mobile robots live in. As a result, many mobile robots need some forms of a map for localization, obstacle

avoidance and navigation. To build such maps, the robots use a predefined data structure, e.g., an occupancy grid [1] or visual features [2], and then perform simultaneous localization and mapping (SLAM). Human beings have a long history of building and using maps. These days one can easily read floor plan maps of airports and shopping centers, and figure out a plan for navigation. By comparison, robots can hardly reach comparable competency in map reading and task planning. It is a non-trivial task for the robots due to the many labels in the map, their ambiguous associations to different areas, and complex geometric configurations. As a result, there is no existing method for addressing the map parsing problem, *i.e.*, computing a navigation plan given a map image and a goal text, which motivated this research.

For complex navigation tasks, a robot needs to compute a task plan, *i.e.*, a sequence of navigation actions, and continuous trajectories for realizing those actions. Example actions can be entering an area and going through a door. Extensive engineering efforts are needed to realize such navigation systems, from building the map itself to labeling areas of the map. At the same time, professional architectural drafters have generated blueprints that accurately reflect the geometric configurations, which unfortunately cannot be used by current robots. From an application perspective, this research aims to leverage the readily accessible floor plan maps in human environments to fulfill the robots' need of maps for navigation.

Vision language models (VLMs) are foundation models that learn from and reason about both images and texts, supporting a variety of downstream tasks from visual question answering to image captioning. VLMs have demonstrated impressive successes in a variety of applications [3–5]. Among them, robotics is an important application domain [6–8]. We, for the first time, apply VLMs to the novel task of map parsing and evaluate its performance in navigation, a foundational problem in mobile robotics. Our approach is simple and intuitive. A floor plan map and a problem description, including the start and goal positions, are provided to a VLM, and the task is to compute a plan (*i.e.*, an action sequence) to achieve the goal. Figure 1 shows a quadruped mobile robot executing a navigation plan that was generated by a VLM using a floor plan map. Despite the straightforward idea, the results are surprisingly good. Navigation plans generated by VLMs which can require a sequence of nine actions are generated correctly up to 90% of the time on some floor plans.

The main contribution of this research includes the introduction of the map parsing problem, evaluations of VLMs on this problem, practical suggestions that paves the way for future research on VLM-based map parsing, and a complete demonstration of real-robot system.

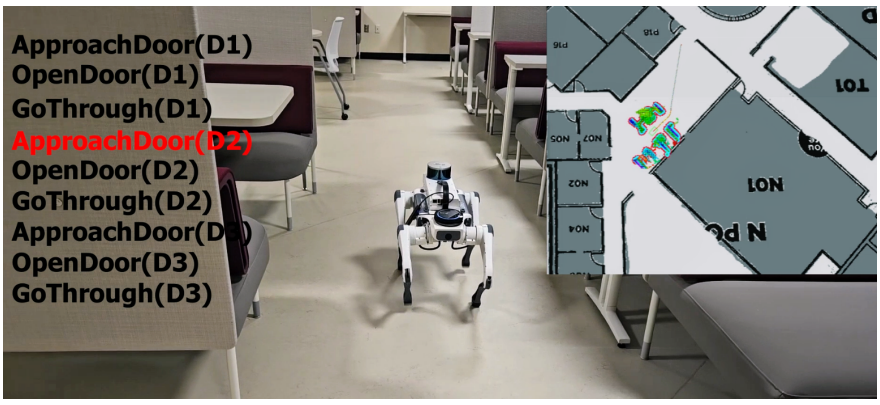


Figure 1. Quadruped robot executing a VLM-generated plan (current action highlighted in red) to complete a navigation task while localizing directly on a floor plan image.

There are limitations in this research that can be addressed in future work. One is that we still need to slightly edit the map image, such as thickening walls and removing architectural annotations, to produce the best performance. Such steps can be automated in future work. Another is that the robot needs to stand close and forward-facing when capturing the map image. This can be a challenge to small robots because most floor plan maps are placed at human heights. In this paper, we focus on highlighting the remarkable performance of VLMs on map parsing, and leave those topics to future work.

2. Related work

In this section, we discuss existing work in autonomous navigation for mobile robots, VLM prompting strategies, and integrating large pre-trained models in robotics. We highlight how our work differs from existing works in each of these categories.

2.1. Existing map representations and navigation

Existing works demonstrating autonomous navigation on mobile robots can be categorized into two groups. One relies on generating an occupancy grid map [9–12], where each pixel in the map identifies an area being occupied, obstacle-free or unknown. Another popular way of realizing mobile navigation leverages vision-based methods for simultaneous localization and mapping (Visual SLAM) [2,13–16]. Those methods rely on visual features, and then apply simultaneously estimate the locations of both visual landmarks and the robot itself [17]. While such map representations have proven to be effective for autonomous navigation tasks, generating an accurate map is oftentimes labor-intensive. For instance, in vision-based settings, the robot has limited knowledge of the global environment, either leading to lengthy exploration, or navigational commands from a human. In this work, we greatly reduce the effort of generating accurate maps while still leveraging detailed information of the environment through the use of existing, and potentially *in situ*, floor plans.

2.2. VLM prompting strategies

The output of a large pre-trained model largely relies on the way it is prompted. Strategies like chain-of-thought [18], in-context learning [19] and others [20,21] are leveraged on LLMs to improve performance. Similar to language prompts, image prompting strategies can improve VLM outputs. Set-of Mark prompting, which segments and labels objects in an image has shown to improve VLM responses [22]. In our work, we design a new visual prompting strategy designed for obtaining a spatial understanding of floor plan images.

2.3. Large models in robotics

To improve the common-sense reasoning capabilities of robots, large pre-trained LLMs and VLMs have been integrated into robotic systems for various tasks. Housekeeping [23,24] and object rearrangement [25–27] are two of the service task examples. Navigation is another robotics domain where LLMs and VLMs have demonstrated exceptional performance [28–32]. Furthermore, while quadruped locomotion problems traditionally were solved using optimization methods, the effectiveness of LLMs and VLMs has been evident in recent years [33–35]. Alignment of the large model with the environment and robot's skills

is critical to perform tasks in the real world [36]. Various approaches have demonstrated planning capabilities [37,38] and uncertainty estimation [39] of large models. Various visual prompting strategies designed for different robotic manipulation tasks have also been developed [6,7]. In line with recent works that leverage large models to incorporate common-sense knowledge on robots, we generate feasible navigation plans directly from an image of a floor plan.

3. Methodology

In this section, we present our approach for leveraging Vision-Language Models (VLMs) to interpret floor plan images and generate navigation instructions. We discuss the two key aspects of our approach: visual prompting strategy, and VLM-based plan generation.

3.1. Visual prompting strategy

Our study uses floor plan images that include detailed architectural layouts for various building types. To generate accurate plans from a VLM, it's important to design a visual prompt which can facilitate learning the structure of the floor plan. However, raw floor plan images tend to contain various markings (*i.e.* windows, furniture symbols, non-uniform wall thickness) which can potentially confuse the VLM in understanding the general layout of the floor plan. Thus, we remove such markings to produce a cleaner map which can be better leveraged by a VLM.

We find that removing extraneous details from the floor plan is insufficient for the VLM to understand the map layout. In particular, we find the VLM has limited spatial awareness for sparsely labelled rooms with lots of open space, and near key decision points like doors and intersections. To alleviate this, we add duplicate room labels in open spaces and near doors and intersections. This provides the VLM with further guidance in understanding the general structure of the floor plan. We later demonstrate the importance of such additional labels in Section 4.

In this study, all floor plans are preprocessed through a simple yet systematic enhancement procedure, next, we present the step-by-step instructions that we followed for map enhancement.

(1) Visual simplification: Raw floor plans frequently include architectural annotations outside the map itself, such as dimensions and scale bars. We remove those visuals and keep only the map part of the floor plans. In addition, there are non structural visuals, such as window icons, and furniture, where people use very different icons or symbols. Those were removed for standardizing evaluations.

(2) Structural standardization: The line segments for walls and doors can be different in floor plan maps. We adjust the wall thickness to a uniform value. Doors are colored yellow, making their locations visually salient and uniformly represented across maps. Windows are replaced with walls to avoid confusing the robots on whether windows are navigable.

(3) Dense labeling: Some rooms are very large, of irregular shape (e.g., long corridors), or both. We added additional labels of rooms at areas near doors or intersections, or within large open areas. This dense labeling provides additional contextual cues, leading to significant performance improvements, especially in complex environments involving multiple rooms and transitions.

(4) Consistency check: The last step is to ensure that all rooms are enclosed, no overlapping or misleading elements remain, and every object (rooms and doors in particular) has a unique ID. The finalized map is then exported as a high-resolution PNG image as the image prompt of the VLMs.

In this work, the map enhancement steps were manually performed, though the process is straightforward and can be reproduced by robotics practitioners using our instructions. We discuss possible ways of automating the map enhancement process near the end of this article as future work.

3.2. VLM-based plan generation

In our VLM-based framework, as shown in Figure 2, VLMs formulate navigation plans based on a floor plan image and a text prompt. This method leverages an instruction-based text prompting strategy, which involves providing the VLM with explicit and detailed guidelines for the navigation task, along with the floor plan image. We now elaborate on the prompting strategy and the resulting output format.

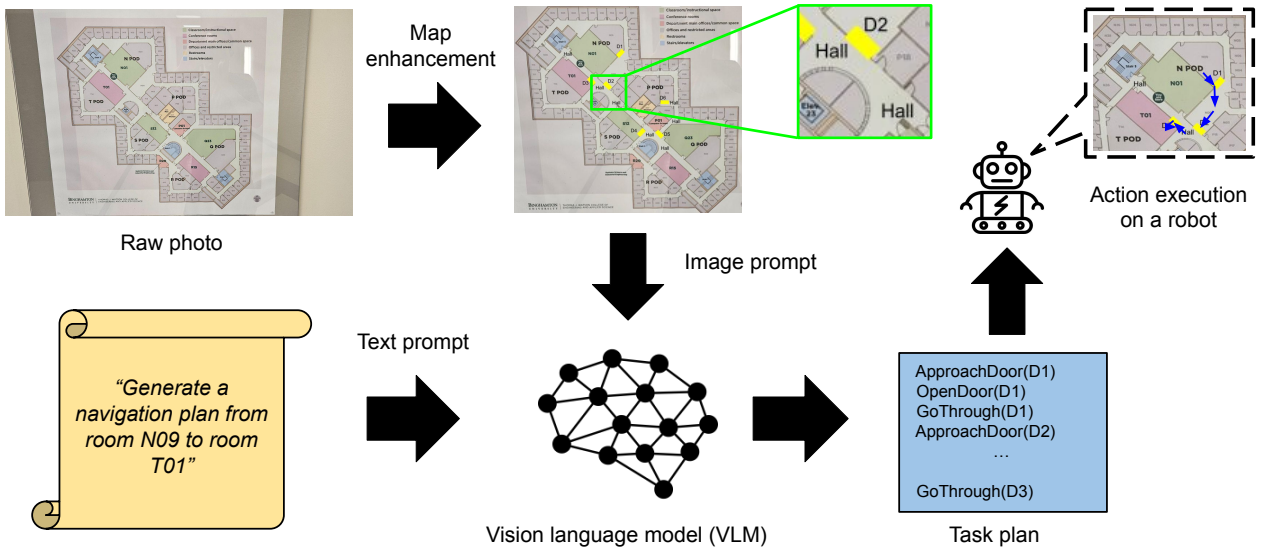


Figure 2. Overview of our method. A robot takes a raw image of a floor plan, which is then enhanced with labels and door indicators. The enhanced floor plan, along with a text prompt specifying the start and goal locations is given to a VLM. The VLM generates a navigation plan to reach the goal location, and the plan is executed on a mobile robot.

The text prompt given to the VLM is shown in Figure 3. The prompt explicitly defines the starting point and the destination. This allows the VLM to understand the required navigation path and objectives clearly. It provides detailed instructions on interpreting the floor plan and managing door interactions. These guidelines include specific protocols for door operations and decision-making processes.

A key aspect of this prompt is the request for all door and room connections. By generating this information at the start, we speculate that the VLM integrates it with the floor plan image to produce an accurate navigation sequence. We believe this step helps the VLM better understand the spatial relationships in the map, leading to accurate navigation path planning.

Based on the text prompt and floor plan image, the VLM generates a sequence of actions required to navigate from the initial to goal location. This sequence includes specific steps, e.g., approaching, opening, and passing through doors.

The output from the VLM is a detailed sequence of actions formatted as follows:

- ApproachDoor(x): Move in front of door x.
- OpenDoor(x): Open door x.
- GoThrough(x): Move through open door x to the location on the other side.

The final navigation plan is output in JSON format, specifying each action. This structured format facilitates easy interpretation and execution of the navigation instructions. This plan is then parsed and executed by the robot.

I am a robot that cannot go through walls and must use doors to navigate. This is the floor plan of the building I am in right now (provided as an image).
You are a navigation agent, and your task is to give me a detailed, efficient navigation plan that strictly follows a sequence of actions to achieve the navigation task: Begin in **<location A>** and arrive at **<location B>**. The only doors which exist are represented as yellow rectangles and labeled with 'D(N)' distinct positive integers(1,2,3...N). A plan consists of a sequence of the following actions:

`ApproachDoor(x)`: Move in front of door x.
`OpenDoor(x)`: Open door x.
`GoThrough(x)`: Move through open door x to the location on the other side.

Include only the necessary doors that are part of the path being used, and do not mention doors that won't be traversed even if they are in the path.

Explicit Room and Door Descriptions: Alongside the image, make a clear list of all rooms and doors with their connections - which is to be used for the navigation task.

Remember that the door symbol can overlap with the boundaries or common spaces. Remember to only use the generated door room connections for making the action plan. Double-check if each action is necessary and correct for traversal to the end goal. Common spaces (eg Hall) and larger rooms may have multiple instances of the same labels to help you understand their boundaries.

Important: The doors close after every `GoThrough(x)` action. Carefully inspect the floor plan image to ensure the correct correspondence between doors and rooms. Prioritize providing a correct path over the shortest path. Make sure the path avoids any unnecessary doors or rooms. If any unnecessary doors or rooms are included, silently correct the plan before providing the final sequence. Give the final path in a json format.

Remember to make explicit connections for each door, then make a step by step solution for each navigation step and ONLY use the door-room connections to generate the final navigation plan.

Figure 3. Text prompt input to VLM to generate navigation plans. We define the starting and ending locations, action types, and ask for explicit room and door connections to gain insights as to how the VLM understands the map.

4. Experiments

In order to evaluate whether the VLM produces accurate navigation plans, we design and run experiments over a dataset of floor plans. The experiments are designed to measure the effect of floor plan size, task difficulty, and label density on the VLM's plan accuracy.

4.1. Experimental setup

Our study uses floor plan images from a publicly available dataset CVC-FP [40], which includes detailed architectural floor plans for various building types. Three floor plans were randomly selected from those that contain 9–11 rooms and clear labels. Those raw maps are shown in Figure 4. As illustrated, these images often contain irrelevant visual elements such as scale bars, dimension lines, and furniture/window symbols, which are not pertinent to the general layout. To facilitate more effective visual prompting for VLMs, we preprocess these raw maps using the procedure described in Section 3.1. The resulting maps are referred to as "original maps" and are shown in Figure 5. Our experiments use two state-of-the-art VLMs: GPT-4o [41] and Claude-3.5 Sonnet [42].

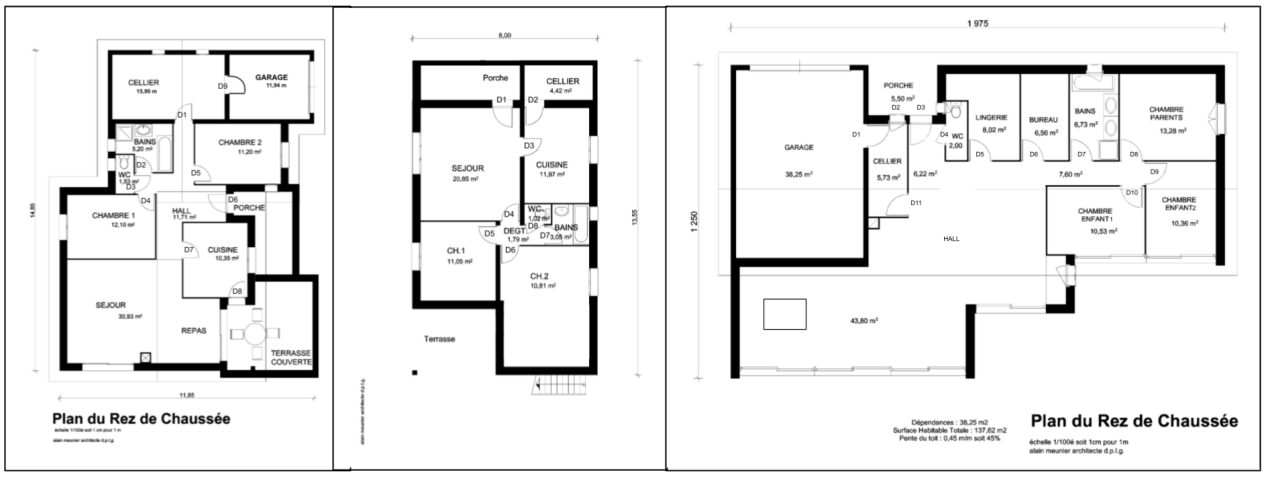


Figure 4. Three raw floor plan maps.

Figure 5. Three maps used in our experiments for evaluating the performance of VLMs in map parsing and plan generation.

For each floor plan, we generate five pairs of start and goal locations. To account for the stochastic nature of VLM responses, we run each VLM ten times per navigation task, resulting in a total of 50 navigation trials per floor plan. We evaluate performance based on the correctness of the generated plans. A plan is considered correct if it uses only those actions defined in the text prompt, includes feasible actions, and leads a sequence of transitions from the start location to the goal. Example infeasible actions

include navigating to a room that is not connected to the current room and opening a door that belongs to a distant room.

Our experimental design focuses on three key dimensions:

(1) Map size: We hypothesize that increasing the map size will result in a decrease in VLM's map parsing performance. This hypothesis is based on the assumption that larger maps introduce greater complexity.

(2) Task difficulty: We hypothesize that when the start and end locations are far away (based on number of rooms required to traverse), the VLMs will have a low accuracy in map parsing and plan generation.

(3) Label density: We hypothesize that a densely labelled floor plan map will facilitate accurate navigation plan generation from VLMs.

Next, we describe our experiment setup for evaluating each of the three hypotheses.

4.1.1. Map size

To examine the impact of map size on navigation performance, we developed two types of maps: Original Maps that are enhanced versions of the maps selected from the CVC-FP dataset, and Doubled Maps that were created by connecting two copies of an original map through an additional door. Figure 6 presents an example of a doubled map. A door denoted as D10 is added to establish connectivity, thereby forming the final doubled map.

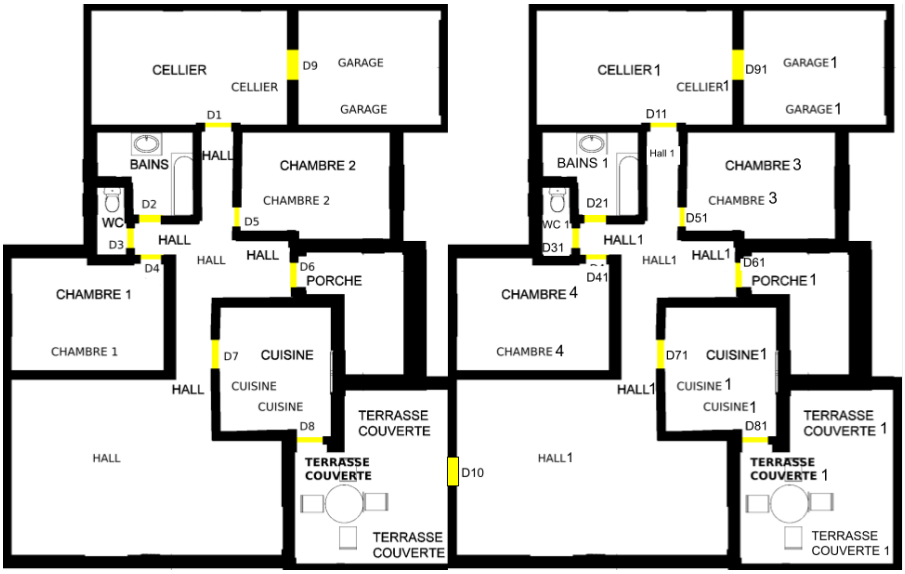


Figure 6. Example of a doubled map.

4.1.2. Task difficulty

We design two types of navigation tasks to evaluate model performance across two levels of difficulty. Easy Tasks are the navigation tasks that can be completed by navigating from Room A to Room B without traversing any intermediate rooms. Hard Tasks are those that require a robot to navigate through at least two intermediate rooms before the goal can be achieved. As a result, an optimal solution of hard tasks will involve four rooms in total. The increased complexity introduces more decision points and possible paths, producing a more challenging task.

4.1.3. Label density

We evaluated the impact of label density on model performance by implementing two labeling schemes. Sparse-Labeled maps are those where each room was labeled with a single label, usually placed at the center. This minimalistic approach offered fewer cues for the model to base its navigation decisions on. Dense-Labeled maps include multiple labels for each room, where the placement is described in Section 3.1.

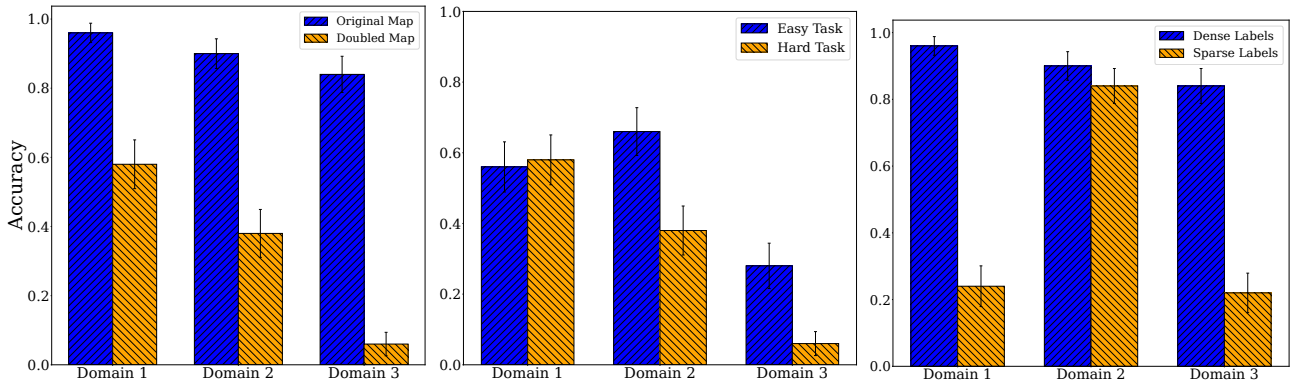
4.2. Experimental results

4.2.1. Hypothesis 1 (map size)

The first hypothesis explores how the size of the map influences the model's accuracy. We compared the performance between original maps and doubled maps over hard tasks to assess the same.

The results indicate that accuracy decreases as map size increases in the overall domain analysis, with the VLMs performing better in smaller maps. The difference is significant, as a T-test on trials with GPT-4o revealed a drop in accuracy ($t = 6.13$, $p < 0.0001$). This supports our hypothesis that accuracy decreases as map size increases. The results are reported in Figures 7 and 8. Both GPT and Claude exhibit a similar pattern of performance decline with larger maps.

When the map size is enlarged, it was observed that the VLM agents frequently had difficulties in figuring out door ownership. For instance, some of the generated plan included infeasible actions such as going through door D from room R, where the door (D) does not belong to room R. This is a strong indication that in large floor plans, it is challenging for the VLMs to reason about door ownership. By comparison, such mistakes were rarely seen in small floor plans.



(a) (b) (c)

Figure 7. Comparison of GPT-4o results across three experimental conditions: **(a)** Map size: performance on original maps *vs.* doubled maps over hard tasks; **(b)** Task difficulty: performance on doubled maps over easy *vs.* hard tasks; **(c)** Label density: performance on original maps annotated with sparse *vs.* dense labels over hard tasks.

We looked into the navigation plans generated by GPT-4o for hard tasks in the three original maps reported in Figure 7, where dense labels were provided. Each map contained five tasks, and each task was run ten times. For instance, Domain 3 included the five navigation tasks of ('GARAGE',

'BUREAU'), ('LINGERIE', 'GARAGE'), ('GARAGE', 'CHAMBRE ENFANT 1'), ('CHAMBRE PARENTS', 'GARAGE'), and ('GARAGE', 'WC'). Among the total 150 trials, GPT-4o was successful in 135 trials by generating navigation plans that correctly connected the initial and goal rooms. Among those 135 successful trials, 132 were optimal. In the 10 trials of the task ('GARAGE', 'WC'), the VLM failed in one trial, and reported suboptimal plan D4–D3–D2–D1 three times, while the optimal path D4–D11–D1 was reported six times. For all other successful trials, the reported plans were optimal.

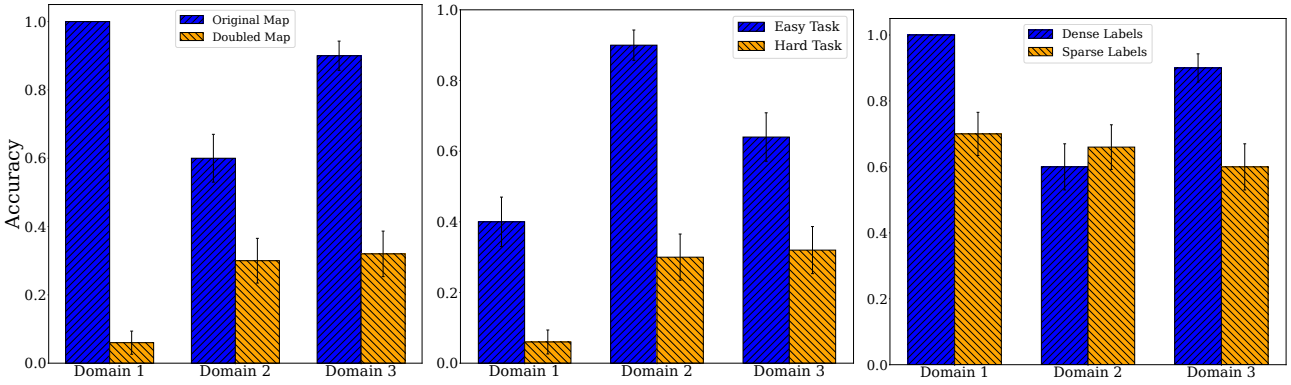


Figure 8. Comparison of Claude Sonnet 3.5 results across three experimental conditions: **(a)** Map Size: performance on original maps *vs.* doubled maps over hard tasks; **(b)** Task Difficulty: performance on doubled maps over easy *vs.* hard tasks; **(c)** Label Density: performance on original maps annotated with sparse *vs.* dense labels over hard tasks.

4.2.2. Hypothesis 2 (task difficulty)

The second hypothesis investigates how task difficulty impacts accuracy. Easy tasks involve straightforward navigation between rooms, while hard tasks, require traversing multiple intermediate rooms, making the tasks more complex. We compared the performance between doubled maps over easy tasks and hard tasks.

The accuracy of the GPT-4o models generally decreased with increased task difficulty, as more complex tasks led to lower accuracy in two out of the three maps. For example, a T-test on doubled_map_2 indicated a drop in accuracy with $t = 2.88$ and $p = 0.0047$. The results from both VLMs generally support our hypothesis that more complex tasks lead to lower accuracy, as navigating through multiple rooms introduces additional challenges. One example task is detailed in Table 1.

Difficult tasks required the VLMs to generate navigation plans to go through three doors, which required strong reasoning capabilities about room connectivity and graph topology. One typical failure case was that VLMs insisted on directly entering a room between the start and goal rooms, when there is no direct access to the room in the middle. Consider the floor plan in Figure 9a. When the robot started in room "N09" and aimed to go to room "Stair 3", VLMs frequently suggested the robot to directly enter room N01. While N01 is immediately adjacent to both N09 and Stair 3, the robot cannot directly enter N01 from N09, because the only door of N01 connects to Hall2 instead of N09. Such observations indicate that VLMs had difficulty in reasoning about topological graphs of room connectivity through doors.

Table 1. An example navigation plan generated by GPT-4o in Map 1 shown on the left of Figure 5. The initial location is "Terrrasse Couverte" and the goal location is "Chambre 1". This navigation task requires a sequence of nine actions. GPT-4o achieved 0.96 success rate in this map on similar hard navigation tasks, which demonstrates great promises for VLM-based map parsing research.

Number	Action
1	ApproachDoor(D8)
2	OpenDoor(D8)
3	GoThrough(D8)
4	ApproachDoor(D7)
5	OpenDoor(D7)
6	GoThrough(D7)
7	ApproachDoor(D4)
8	OpenDoor(D4)
9	GoThrough(D4)

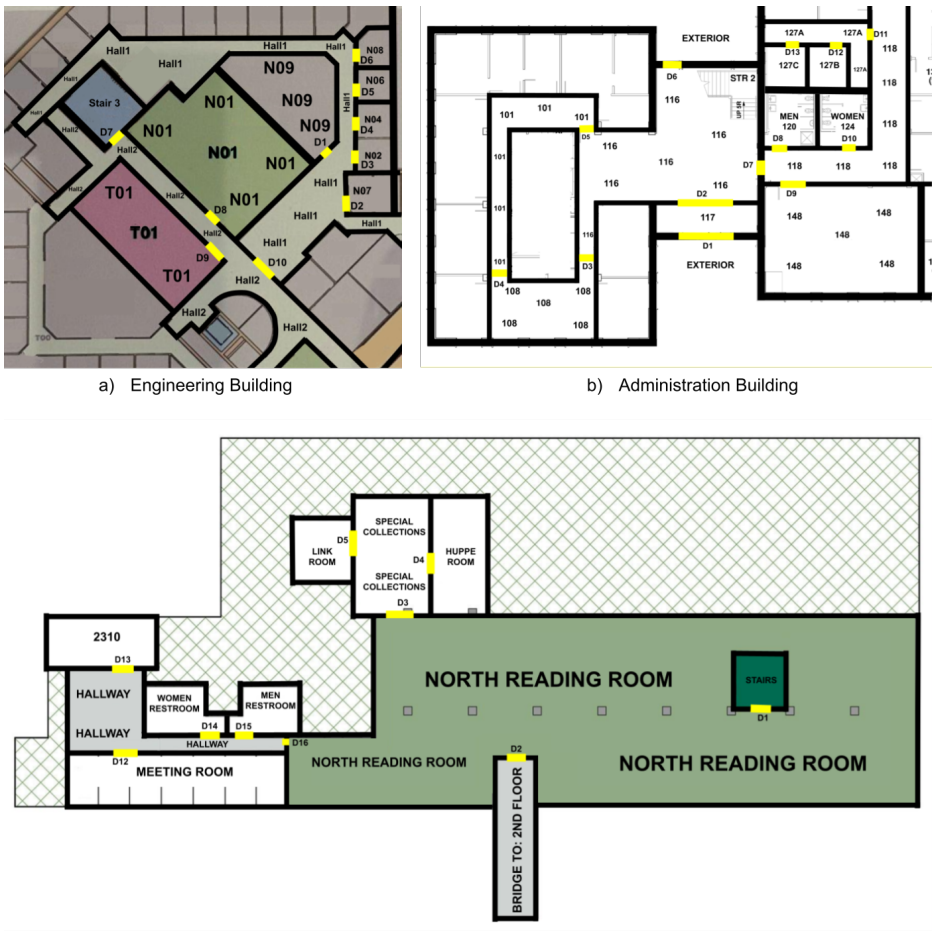


Figure 9. Three additional maps from the college campus. **(a)** Engineering Building; **(b)** Administration Building; **(c)** Library.

4.2.3. Hypothesis 3 (label density)

The third hypothesis examines the impact of label density on accuracy. This is evaluated by comparing the performance of original maps from sparse-label and dense-label datasets over hard tasks. Dense labels provide more contextual information, while sparse labels offer minimal context, making the navigation task more challenging.

The accuracy of the GPT-4o models significantly improved with dense labels compared to sparse labels across all maps. For instance, a T-test on original_map_1 showed a significant improvement with $t = 10.72$ and $p < 0.0001$. This result indicates that dense labels substantially enhance accuracy, supporting our hypothesis that dense labels provide crucial contextual information, enabling the models to navigate more effectively. The improvement in accuracy with dense labels was consistent across all maps for GPT-4o, highlighting the importance of label density in successful navigation.

Without dense labeling, VLMs frequently made mistakes in reasoning about door ownership and room connectivity through doors. For instance, door D16 in Figure 9b connects two rooms "NORTH READING ROOM" and "HALLWAY", where the former is very large. We found that the VLMs sometimes wrongly believed that D16 connects "MEN RESTROOM" and "HALLWAY" because those two room labels are closer to the door label. To facilitate VLMs to reason about such door ownerships, we added two additional labels for the reading room, and one of them was placed next to door D16, where dense labeling was found useful for map parsing and navigation task planning.

These findings suggest that our approach can effectively handle a range of navigation challenges while demonstrating the critical importance of label density, task difficulty, and map size in determining overall performance.

4.2.4. Raw *vs.* processed floor plan maps

We evaluated VLM performance on raw floor plan maps for the hard tasks. As shown in Figure 4, these raw images contain various irrelevant elements that are not essential to the spatial layout, such as furniture symbols, dimension lines, and scale bars. The results, presented in Table 2, show that accuracy is substantially lower when using raw maps compared to the processed sparse-label maps. This supports our claim in Section 3.1 that raw maps contain distracting visual elements which may confuse the VLM in understanding the overall layout.

Table 2. Average accuracy results comparing processed and raw floor plan maps across three domains for hard tasks.

Map Type	GPT-4o			Claude Sonnet 3.5		
	Domain 1	Domain 2	Domain 3	Domain 1	Domain 2	Domain 3
Processed Maps	0.24	0.84	0.22	0.68	0.66	0.34
Raw Maps	0.03	0.21	0.05	0.08	0.18	0.22

4.2.5. Diverse floor plans

To evaluate the generalizability of our conclusions across different environments, we extended our experiments to three additional floor plans selected from the home institution of this article's authoring team, including the Engineering Building, the Library, and the Administration Building. The Engineering Building is also the environment where we conducted the real-world demonstration, described in Section 5 on Hardware Demonstration. These three floor plans are illustrated in Figure 9. The results, presented in Table 3, show that the VLMs performed well on the processed maps.

Table 3. Average accuracy of VLM performance on three college campus floor plans.

Task Difficulty	GPT-4o			Claude Sonnet 3.5		
	Engineering	Administration	Library	Engineering	Administration	Library
Easy Task	0.92	0.90	0.92	0.78	0.70	0.94
Hard Task	0.80	0.94	0.90	0.54	0.62	0.90

5. Hardware demonstration

Our VLM-based planning and navigation system is demonstrated on a DEEP Robotics Lite3 quadruped robot. An image of the floor plan of a building on a college campus is captured from the robot's camera. This image is then edited as described in Section 3 to make it suitable for the VLM query. Another version of the raw image with space whited out is directly used for robot localization, as seen in Figure 10. The VLM is then queried with the edited photo, and a navigation plan consisting of a sequence of navigation and door opening actions is generated. The robot executes this navigation plan, to move from the robotics lab (room N09), to a classroom (room T01). The robot localizes itself directly on a grayscale version of the floor plan, without the need for generating an accurate occupancy grid via SLAM.

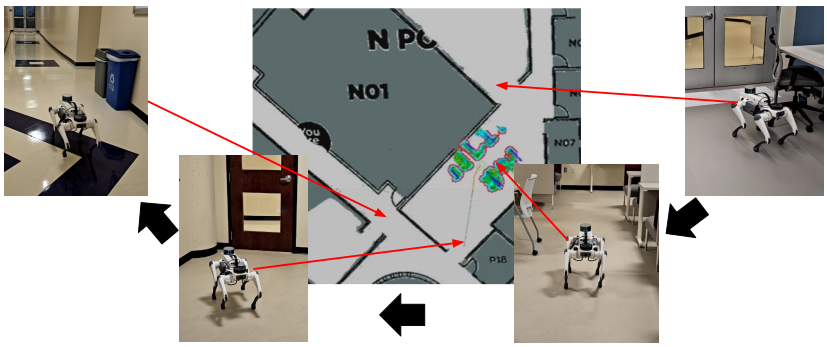


Figure 10. The robot localizes directly on the floor plan image and performs a navigation task. The occupancy grid map that the robot used for localization and navigation was also derived from the floor plan map captured by the robot. We manually removed the labels that would otherwise be interpreted by the robot as obstacles. From right to left, the robot was performing (1) Going through Door N09, (2) Approaching Door N00, (3) Opening Door N00, and (4) Approaching Door T01.

The robot's software for localization, obstacle avoidance and navigation, including actions of

approaching and going through doors, was constructed using an existing software codebase for building-wide robot intelligence [43]. The codebase supported labeling door areas on both sides of a door. Whenever the robot gets inside either of the door areas, it is believed that the robot successfully completes the action of approaching the door. The action of going through a door is realized by moving from one of the door areas to the other side of the same door. Our robot cannot open doors by itself and door opening actions are realized by repeatedly saying "Please help me open the door" to seek human help until the two areas of the door become accessible to each other. We used the Lite3 robot's locomotion skills that rely on its default Model Predictive Controller (MPC). The robot used standard methods from the ROS community for navigation, including the dynamic window approach (DWA) for obstacle avoidance, A* for global path planning, and Adaptive Monte Carlo Localization (AMCL) for localization using grayscale occupancy-grid map images. The robot successfully avoided the obstacles, asked for doors to be opened as needed, and achieved the desired navigation goal, as illustrated in Figure 10.

6. Conclusion and future work

In this work, we introduce a novel task, named map parsing, that is unexplored in the VLM literature while pointing to the foundation of mobile robotics. We demonstrate remarkable performance of two VLMs on map parsing tasks, as applied to robot navigation. We develop a VLM-based planning system that generates navigation plans directly from a floor plan image and validated our approach through experiments on a floor plan dataset and on hardware, demonstrating the feasibility of VLM-driven navigation across different environments.

While our results show that this approach is viable, several challenges remain, opening up avenues for further research. While the process of floor map enhancement was manually done in this article, one can easily automate the process using the following steps to reproduce our results. First, all labels and text outside the floor plans should be removed. One can use segmentation models [44–46] to locate those items for removal. After that, the formats of walls including thickness and line style should be unified. For instance, walls with windows are frequently indicated using hollow boxes, and it's necessary to replace them with solid line segments. After that, all door icons should be unified. While we did not perform a comprehensive comparison, yellow rectangles were observed effective in our implementation. Since the navigation plans frequently include actions of going through doors [47], it is necessary to give each door a unique ID by adding a unique door name next to each door icon in the floor plans. Finally, dense labeling of rooms is necessary by copying and pasting room names to open and near-door areas of large rooms.

Another direction for automating the floor map enhancement pipeline is to leverage the state-of-the-art image editing tools. For instance, one can prompt ChatGPT [22] or Gemini [48] with the raw map as image prompt and with editing instructions as the text prompt (e.g., replacing door symbols in this floor plan with solid yellow rectangles). To further improve the performance, one can employ chain-of-thought (CoT) [18] style reasoning along with few-shot examples to prompt the model to follow the manual sequence of operations. Those operations include removing external labels and annotations, normalizing wall and window representations, standardizing door icons, and assigning unique IDs to each door. An alternative way of automating the map enhancement process is to leverage segmentation tools

such as SAM 2 [45] to mark out objects in the floor map and assign symbolic labels to the identified objects. Accordingly, walls can be thickened and door colors can be changed automatically, as a follow-up step. Automating the process of map enhancement can be an interesting and important future work direction to enhance this research from the engineering perspective.

Finally, researchers can investigate strategies to improve the VLM's ability to handle larger and more complex maps, e.g., outdoor environments [49]. Public parks and college campuses have been studied as the playground of mobile robot navigation [50,51], and they are usually provide visitors with maps near the entrances. Those maps have rich labels and spatial information. While we only evaluated our approach in indoor environments in this paper, we believe our approach can be furthered towards outdoor environments with larger areas and complex structures. VLMs and other transformer based autoregressive models have a host of known issues, such as hallucinations and biases [52–55] that can be addressed in robot planning, which deserves more attention too. For instance, VLMs might assume every home environment has a kitchen (which is wrong in places) [56] and wrongly suggest a robot to go to a kitchen to get water in the kitchen that does not exist. To really bring our map parsing approach and the robot systems to end users, it is necessary to develop safeguards to ensure the output of VLMs do not introduce risks to people and the environments [57,58]. This is particularly important when mobile robots are used for guiding people who are visually or physically challenged, e.g., robotic guide dogs [12,59,60], where this work was initially motivated by such application domains. We anticipate that this work will inspire further studies that expand upon the ideas on VLM-based map parsing in this paper.

Data availability statement

The data or datasets generated or analyzed in this study are available in GitHub, where the link is available on the project webpage: https://sites.google.com/view/vlm-floorplan/.

Acknowledgments

A portion of this work has taken place at the Autonomous Intelligent Robotics (AIR) Group, SUNY Binghamton. AIR research is supported in part by the NSF (IIS-2428998, NRI-1925044), Ford Motor Company, DEEP Robotics, OPPO, Guiding Eyes for the Blind, and SUNY RF.

Authors' contribution

DeFazio contributed to Methodology, Data curation, Formal analysis, Writing—review and editing, Software, Conceptualization, Investigation, Writing—original draft. Mehta contributed to Methodology, Data curation, Formal analysis, Writing—review & editing, Software, Conceptualization, Investigation, Writing–original draft. Wang contributed to Methodology, Data curation, Formal analysis, Writing—review & editing, Software, Conceptualization, Investigation, Writing—original draft. Yang contributed to Methodology, Data curation, Formal analysis, Writing—review & editing, Funding acquisition, Supervision, Conceptualization, Writing–original draft. Blackburn contributed to Methodology, Data curation, Formal analysis, Writing—review & editing, Funding acquisition, Supervision, Conceptualization, Writing–original draft. Zhang contributed to Methodology, Data curation,

Formal analysis, Writing—review & editing, Funding acquisition, Supervision, Conceptualization, Writing—original draft.

Conflicts of interests

The authors declare no conflict of interest.

References

[1] Elfes A. Using occupancy grids for mobile robot perception and navigation. *Computer* 1989, 22(6):46–57.

[2] Taketomi T, Uchiyama H, Ikeda S. Visual SLAM algorithms: a survey from 2010 to 2016. *IPSJ Trans. Comput. Vis. Appl.* 2017, 9:1–11.

[3] Antol S, Agrawal A, Lu J, Mitchell M, Batra D, *et al.* Vqa: visual question answering. In *Proceedings of the IEEE international conference on computer vision*, Santiago, Chile, December 7–13, 2015, pp. 2425–2433.

[4] Chen Y, Li L, Yu L, El Kholy A, Ahmed F, *et al.* Uniter: universal image-text representation learning. In *European conference on computer vision*, Munich, Germany, August 24–27, 2020, pp. 104–120.

[5] Brooks T, Holynski A, Efros AA. Instructpix2pix: learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Vancouver, Canada, June 18–22, 2023, pp. 18392–18402.

[6] Nasiriany S, Xia F, Yu W, Xiao T, Liang J, *et al.* PIVOT: iterative visual prompting elicits actionable knowledge for VLMs. *arXiv* 2024, arXiv:2402.07872.

[7] Liu F, Fang K, Abbeel P, Levine S. MOKA: open-vocabulary robotic manipulation through mark-based visual prompting. *arXiv* 2024, arXiv:2403.03174.

[8] Hu Y, Xie Q, Jain V, Francis J, Patrikar J, *et al.* Toward general-purpose robots via foundation models: A survey and meta-analysis. *arXiv* 2023, arXiv:2312.08782.

[9] Zhang S, Yang F, Khandelwal P, Stone P. Mobile robot planning using action language with an abstraction hierarchy. In *International Conference on Logic Programming and Nonmonotonic Reasoning*, Lexington, USA, September 27–30, 2015, pp. 502–516.

[10] Hayamizu Y, Amiri S, Chandan K, Takadama K, Zhang S. Guiding robot exploration in reinforcement learning via automated planning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, Vienna, Austria, July 11–15, 2021, pp. 625–633.

[11] Fu Z, Kumar A, Agarwal A, Qi H, Malik J, *et al.* Coupling vision and proprioception for navigation of legged robots. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, USA, June 19–20, 2022, pp. 17273–17283.

[12] DeFazio D, Hirota E, Zhang S. Seeing-eye quadruped navigation with force responsive locomotion control. In *Conference on Robot Learning*, Atlanta, USA, November 6–9, 2023, pp. 2184–2194.

[13] Zhang C, Yang Z, Fang Q, Xu C, Xu H, *et al.* FRL-SLAM: a fast, robust and lightweight SLAM system for quadruped robot navigation. In *2021 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Sanya, China, December 27–31, 2021, pp. 1165–1170.

[14] Macario Barros A, Michel M, Moline Y, Corre G, Carrel F. A comprehensive survey of visual slam algorithms. *Robotics* 2022, 11(1):24.

[15] Sorokin M, Tan J, Liu CK, Ha S. Learning to navigate sidewalks in outdoor environments. *IEEE Rob. Autom. Lett.* 2022, 7(2):3906–3913.

[16] Hwang H, Xia T, Keita I, Suzuki K, Biswas J, *et al.* System configuration and navigation of a guide dog robot: toward animal guide dog-level guiding work. In *IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, May 29–June 2, 2023, pp. 9778–9784.

[17] Thrun S. Probabilistic robotics. *Communications of the ACM* 2002 45(3):52–57.

[18] Wei J, Wang X, Schuurmans D, Bosma M, Xia F, *et al.* Chain-of-thought prompting elicits reasoning in large language models. *Adv. Neural Inf. Process. Syst.* 2022, 35:24824–24837.

[19] Dong Q, Li L, Dai D, Zheng C, Wu Z, *et al.* A survey on in-context learning. *arXiv* 2022, arXiv:2301.00234.

[20] Yao S, Yu D, Zhao J, Shafran I, Griffiths T, *et al.* Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems* 2023, 36:11809–11822.

[21] Besta M, Blach N, Kubicek A, Gerstenberger R, Podstawski M, *et al.* Graph of thoughts: solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, Vancouver, Canada, February 20–27, 2024, pp. 17682–17690.

[22] Yang J, Zhang H, Li F, Zou X, Li C, *et al.* Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv* 2023, arXiv:2310.11441.

[23] Kant Y, Ramachandran A, Yenamandra S, Gilitschenski I, Batra D, *et al.* Housekeep: tidying virtual households using commonsense reasoning. In *European Conference on Computer Vision*, Tel Aviv, Israel, October 23–27, 2022, pp. 355–373.

[24] Li C, Zhang R, Wong J, Gokmen C, Srivastava S, *et al.* Behavior-1k: a benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, Atlanta, USA, November 6–9, 2023, pp. 80–93.

[25] Ding Y, Zhang X, Paxton C, Zhang S. Task and motion planning with large language models for object rearrangement. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Detroit, USA, October 1–5, 2023, pp. 2086–2092.

[26] Huang E, Jia Z, Mason MT. Large-scale multi-object rearrangement. In *2019 international conference on robotics and automation (ICRA)*, Montreal, Canada, May 20–24, 2019, pp. 211–218.

[27] Danielczuk M, Mousavian A, Eppner C, Fox D. Object rearrangement using learned implicit collision functions. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China, May 30–June 5, 2021, pp. 6010–6017.

[28] Yokoyama N, Ha S, Batra D, Wang J, Bucher B. Vlfm: vision-language frontier maps for zero-shot semantic navigation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, Yokohama, Japan, May 13–17, 2024, pp. 42–48.

[29] Rajvanshi A, Sikka K, Lin X, Lee B, Chiu HP, *et al.* Saynav: grounding large language models for dynamic planning to navigation in new environments. In *Proceedings of the International Conference on Automated Planning and Scheduling*, Banff, Canada, June 1–6, 2024, pp. 464–474.

[30] Song D, Liang J, Payandeh A, Xiao X, Manocha D. Socially aware robot navigation through scoring

using vision-language models. *arXiv* 2024, arXiv:2404.00210.

[31] Sathyamoorthy AJ, Weerakoon K, Elnoor M, Zore A, Ichter B, *et al.* CoNVOI: context-aware navigation using vision language models in outdoor and indoor environments. *arXiv* 2024, arXiv:2403.15637.

[32] Chen AS, Lessing AM, Tang A, Chada G, Smith L, *et al.* Commonsense reasoning for legged robot adaptation with vision-language models. *arXiv* 2024, arXiv:2407.02666.

[33] Yu W, Gileadi N, Fu C, Kirmani S, Lee KH, *et al.* Language to rewards for robotic skill synthesis. *arXiv* 2023, arXiv:2306.08647.

[34] Tang Y, Yu W, Tan J, Zen H, Faust A, *et al.* Saytap: language to quadrupedal locomotion. *arXiv* 2023, arXiv:2306.07580.

[35] Gu Z, Li J, Shen W, Yu W, Xie Z, *et al.* Humanoid locomotion and manipulation: current progress and challenges in control, planning, and learning. *arXiv* 2025, arXiv:2501.02116.

[36] Ahn M, Brohan A, Brown N, Chebotar Y, Cortes O, *et al.* Do as I can, not as I say: grounding language in robotic affordances. *arXiv* 2022, arXiv:2204.01691.

[37] Huang W, Xia F, Xiao T, Chan H, Liang J, *et al.* Inner monologue: embodied reasoning through planning with language models. *arXiv* 2022, arXiv:2207.05608.

[38] Liu B, Jiang Y, Zhang X, Liu Q, Zhang S, *et al.* Llm+ p: empowering large language models with optimal planning proficiency. *arXiv* 2023, arXiv:2304.11477.

[39] Ren AZ, Dixit A, Bodrova A, Singh S, Tu S, *et al.* Robots that ask for help: uncertainty alignment for large language model planners. *arXiv* 2023, arXiv:2307.01928.

[40] De las Heras LP, Terrades O, Robles S, S'anchez G. CVC-FP and SGT: a new database for structural floor plan analysis and its groundtruthing tool. *Int. J. Doc. Anal. Recogn.* 2015, 18:15–30.

[41] OpenAI. GPT-4o. Available: https://openai.com/index/hello-gpt-4o/ (Accessed on 20 May 2024).

[42] Anthropic. Claude-3.5 sonnet. Available: https://www.anthropic.com/news/claude-3-5-sonnet (Accessed on 13 September 2024).

[43] Khandelwal P, Zhang S, Sinapov J, Leonetti M, Thomason J, *et al.* Bwibots: a platform for bridging the gap between ai and human–robot interaction research. *Int. J. Rob. Res.* 2017, 36(5–7):635–659.

[44] Zhang C, Han D, Qiao Y, Kim JU, Bae SH, *et al.* Faster segment anything: towards lightweight sam for mobile applications. *arXiv* 2023, arXiv:2306.14289.

[45] Ravi N, Gabeur V, Hu Y, Hu R, Ryali C, *et al.* Sam 2: segment anything in images and videos. *arXiv* 2024, arXiv:2408.00714.

[46] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, Boston, USA, June 8–10, 2015, pp. 3431–3440.

[47] Thomaz A. Robots in real life: putting hri to work. In *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, Stockholm, Sweden, March 13–16, 2023, p. 3.

[48] Comanici G, Bieber E, Schaekermann M, Pasupat I, Sachdeva N, *et al.* Gemini 2.5: pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv* 2025, arXiv:2507.06261.

[49] Shah D, Osiński B, Levine S. LM-Nav: robotic navigation with large pre-trained models of language, vision, and action. In *Proceedings of the 6th Conference on Robot Learning*, Atlanta, USA, December

14–18, 2023, pp. 492–504.

[50] Karnan H, Nair A, Xiao X, Warnell G, Pirk S, *et al.* Socially compliant navigation dataset (scand): a large-scale dataset of demonstrations for social navigation. *IEEE Rob. Autom. Lett.* 2022, 7(4):11807–11814.

[51] Song D, Liang J, Payandeh A, Raj AH, Xiao X, *et al.* Vlm-social-nav: socially aware robot navigation through scoring using vision-language models. *IEEE Rob. Autom. Lett.* 2024, 10:508–515.

[52] Zhang Y, Li Y, Cui L, Cai D, Liu L, *et al.* Siren's song in the AI ocean: a survey on hallucination in large language models. *arXiv* 2023, arXiv:2309.01219.

[53] Bender EM, Gebru T, McMillan-Major A, Shmitchell S. On the dangers of stochastic parrots: can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, Canada, March 3–10, 2021, pp. 610–623.

[54] Si WM, Backes M, Blackburn J, De Cristofaro E, Stringhini G, *et al.* Why so toxic? Measuring and triggering toxic behavior in open-domain chatbots. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, Los Angeles, USA, November 7–11, 2022, pp. 2659–2673.

[55] Garg S, Ramakrishnan G. BAE: BERT-based adversarial examples for text classification. *arXiv* 2020, arXiv:2004.01970.

[56] Wu Y, Wu Y, Tamar A, Russell S, Gkioxari G, *et al.* Bayesian relational memory for semantic visual navigation. In *Proceedings of the IEEE/CVF international conference on computer vision*, Seoul, Republic of Korea, October 27–November 2, 2019, pp. 2769–2779.

[57] Ravichandran Z, Robey A, Kumar V, Pappas GJ, Hassani H. Safety Guardrails for LLM-Enabled Robots. *arXiv* 2025, arXiv:2503.07885.

[58] Robey A, Ravichandran Z, Kumar V, Hassani H, Pappas GJ. Jailbreaking llm-controlled robots. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, Atlanta, USA, May 19–23, 2025, pp. 11948–11956.

[59] Kim JT, Byrd M, Crandell JL, Walker BN, Turk G, *et al.* Understanding expectations for a robotic guide dog for visually impaired people. In *2025 20th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Melbourne, Australia, March 3–6, 2025, pp. 262–271.

[60] Hwang H, Jung HT, Giudice NA, Biswas J, Lee SI, *et al.* Towards robotic companions: understanding handler-guide dog interactions for informed guide dog robot design. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, Honolulu, USA, May 11–16, 2024, pp. 1–20.