Vision-Language Based Semantic Navigation in Construction

Saika Wong[1], Shaobin Zhou[1], Zihan Zhou[2], and Mi Pan[1]

[1]Department of Civil and Environmental Engineering, University of Macau, Macau SAR, China
[2]Department of Engineering, University of Cambridge, Cambridge, United Kingdom

sk.wong@um.edu.mo, shaobin.zhou@um.edu.mo, zz535@cam.ac.uk, mipan@um.edu.mo

Abstract -

Robotic navigation in construction environments remains challenging due to unstructured layouts, frequent environmental changes, and weak semantic regularities, which limit the applicability of current navigation approaches. Recent vision–language models offer new opportunities for semantic goal-oriented navigation by enabling robots to reason over high-level language instructions and visual observations. However, their effectiveness in construction settings remains underexplored. This paper explores the feasibility and limitations of vision–language–based semantic navigation in unstructured indoor construction environments. We adopt a modular navigation pipeline and evaluate it in a custom simulated indoor construction site. Through the controlled experiment, we analyze navigation performance under different perception and exploration paradigms and identify dominant failure modes related to object detection and semantic guidance. The results indicate that while stronger perception models significantly improve the success rate and detection reliability, overall navigation efficiency remains constrained by weak semantic cues inherent to construction environments. These findings highlight both the potential and current limitations of vision–language–based navigation in construction, and provide insights for future research on robust semantic navigation under unstructured and semi-structured conditions.

Keywords -

Construction robot; Semantic navigation; Goal-oriented navigation; Vision language model

1 Introduction

Robotic systems have been widely deployed to improve productivity, safety, and automation in construction environments. They support a range of applications such as on-site waste sorting [8], object inspection [15], energy auditing [4], and data collection [3]. However, unlike industrial or household settings, construction environments are inherently unstructured and complex. Typical conditions include frequently changing object layouts, random placement of materials and equipment, partially occluded objects. These characteristics pose significant challenges for robotic navigation and limit the applicability of methods that rely on structured geometry information or prebuilt maps [3].

Current navigation approaches in construction mainly rely on geometry-driven [1] and map-based methods [2], for which navigation is formulated as a geometric problem under the assumption of well-defined environments and focusing on accurate localization and collision-free path planning. This paradigm is restrictive in real-world construction practice, where navigation is driven by high-level, human-specified tasks rather than predefined coordinates. In scenarios such as object inspection [4], robots are commonly required to locate objects without access to accurate prior maps and are instructed using high-level task descriptions that refer to object categories, functional roles, and spatial relations. These requirements necessitate the navigation systems to establish explicit semantic grounding between task descriptions and the environment under conditions of limited prior knowledge.

Recent advances in vision–language models (VLMs) have shown strong potential in bridging visual perception and high-level semantic reasoning [10]. By aligning visual observations with natural language descriptions, these models enable robots to reason the relationships between objects and scenes [14]. Such capabilities are particularly appealing for the above-mentioned construction activities. While vision–language–based methods have been extensively studied in household or office environments [21], their effectiveness and limitations in construction environments, characterized by weak semantic regularities and more complex visual conditions, remain largely unexplored. In particular, it remains unclear how perception quality influences overall navigation performance when high-level semantic guidance is employed in such environments.

The paper aims to investigate the feasibility of vision-language-based methods for navigation in unstructured construction environments. Specifically, we explore how existing vision-language navigation pipelines perform and identify the factors that influence navigation outcomes when deployed in a construction context. To this end, we build a simulated indoor construction-site environment that captures key characteristics of real-world construction

settings, including diverse object categories, cluttered layouts, and irregular object placement. Drawing on the framework proposed by Yokoyama et.al [19], we target the navigation task of this study as Object Goal Navigation (ObjectNav), in which a robot aims to search for an instance of a specified object category in a previously unseen environment. We then conduct controlled experiments by varying both the perception models and the exploration strategies, including a frontier-based exploration without vision-language guidance, to evaluate their effects on navigation success and efficiency.

2 Related work

2.1 Robot Navigation in Construction

Compared to robot navigation in general indoor or structured settings, navigating in construction environments poses challenges due to the nature of highly dynamic and continuously evolving, with temporary equipment and materials that invalidate static or long-term maps.

There are two dominant navigation paradigms applied in construction: geometry-driven [1] and map-based navigation [2]. The first paradigm utilizes LiDAR- or RGB-D-based SLAM integrated with classical path planning, which often assumes static or slowly changing environments. The second paradigm typically uses BIM as the prior map to facilitate localization and navigation. Both methods inevitably encounter issues, such as requiring consistent spatial layouts, being sensitive to deviations between planned and as-built conditions, and being unable to reason about semantic concepts relevant to navigation tasks. In addition, the ultimate navigation objectives in construction are typically task-driven [15, 4], such as reaching a specific functional area or object of interest, rather than precise geometric coordinates. In this regard, navigation goals are commonly specified at a high semantic level (e.g., around bricks pile), requiring robots to interpret contextual cues in the environment that go beyond purely geometric representations.

2.2 Semantic Goal-Oriented Navigation

Semantic navigation has emerged as a promising approach to improve the reasoning and generalization capabilities of robot navigation, in which goals can be defined by object [7] or vision-language instructions [14]. More specifically, studies show that learning from semantic representations and optimizing image-text similarity across diverse environments can enhance agents' ability to generalize to unseen settings [4, 3]. These improvements can alleviate the dependency on precise maps, increase robustness to layout changes, and enable more task-aligned navigation decisions, resulting in visually and semantically grounded navigation behaviors [21, 19]. Despite these advances, semantic goal-oriented navigation has predominantly been studied in household or office environments, where semantic features are well structured spatially [21, 19, 7, 13]. The application of such methods to an unstructured or semi-structured environment, e.g., construction site, remains underexplored, leaving potential untapped for leveraging visual and semantic cues in robot navigation within construction contexts.

2.3 Application of Foundation Models in Construction

Early studies on robotic navigation in construction predominantly relied on task-specific perception models trained on predefined target categories [8, 9]. In these approaches, perception was primarily responsible for object detection, whereas navigation and path planning were conducted using geometry-based methods such as A* or Dijkstra. With the emergence of foundation models, recent research has begun to integrate these models into construction automation. However, applications mainly focused on vision–language understanding tasks, such as visual question answering [15] and semantic querying [6], where foundation models were used to interpret visual scenes or respond to textual queries. Only a few studies have explored the use of foundation models to actively guide navigation in construction settings, such as Cai et al. [4] incorporated semantic cues through heuristic frontier-based mechanisms. This gap highlights the disconnect between current navigation paradigms and the requirements of semantic, adaptable navigation in unstructured construction environments.

3 Method

3.1 Problem Formulation

In this work, we address the problem of ObjectNav for robotic agents in unstructured construction environments. The task of ObjectNav is defined as requiring an agent to search for a specified target object by exploring and navigating in a previously unseen environment, i.e., without any task-specific training or pre-built maps. As this semantic navigation exploits high-level semantic concepts rather than pure geometric features, at each timestep t, the agent only receives egocentric RGB-D observations $I_t = \left(I_t^{\text{RGB}}, I_t^{\text{DEPTH}}\right)$, where $I_t^{\text{RGB}} \in \mathbb{R}^{H\times W\times 3}$ and $I_t^{\text{DEPTH}} \in \mathbb{R}^{H\times W}$, its pose $p_t = (x_t, R_t)$ from the odometry sensor, where $x_t = (x_t, y_t, z_t)$ and $R_t \in \text{SO}(3)$, and a specified target object o in the form of natural language. The agent then outputs an action $a_t \in \mathcal{A}$. The action space $\mathcal{A}$ contains four discrete actions: Move Forward (0.25 m), Turn Left (30°), Turn Right (30°), and Stop. An episode is considered successful when the agent issues the Stop action within 500 steps while being within

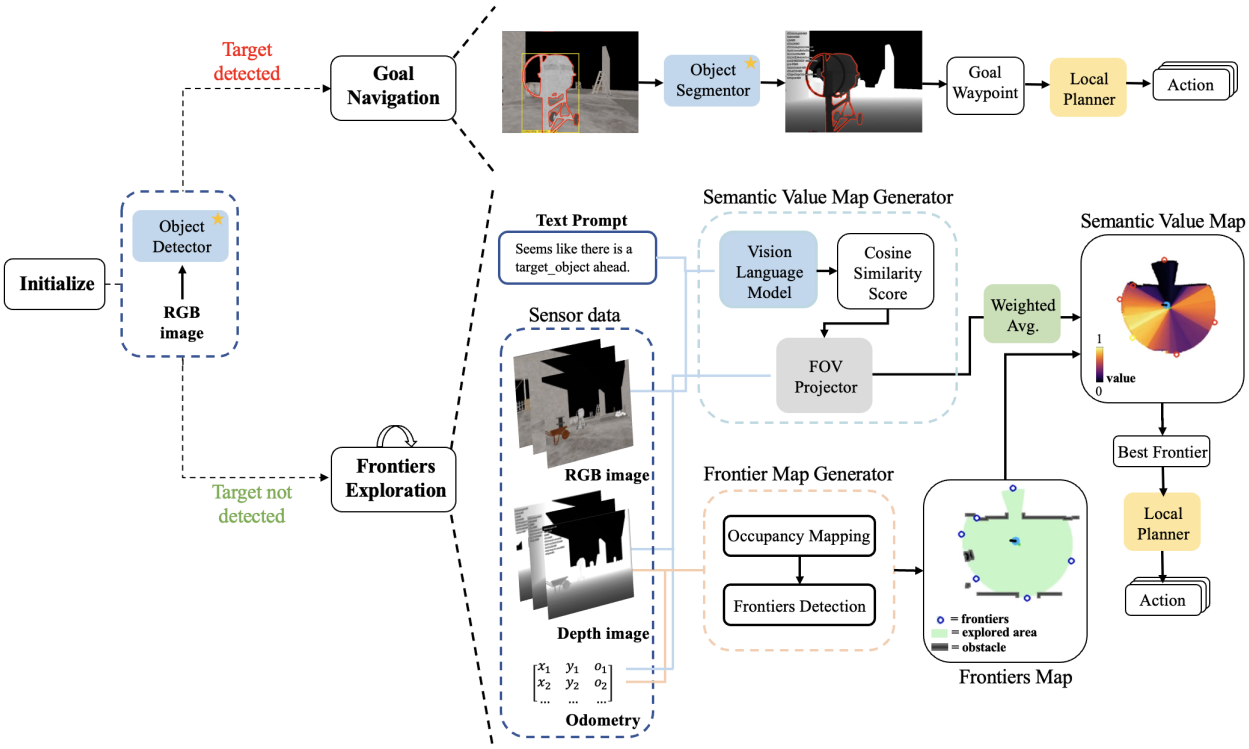

Figure 1. **Overview of the vision-language-based navigation pipeline.** The overall framework follows the VLFM [19] architecture, with the perception models highlighted as the experiment variable. The agent alternates between frontiers exploration mode and goal navigation mode based on whether target is detected or not.

a success radius (less than 1 m) of any instance of the target object.

3.2 Vision-Language-based Navigation Pipeline

Overview. As depicted in Figure 3, the agent initializes the navigation process by turning 360° to acquire a coarse understanding of the surrounding environment. The detector is operated in parallel based on the acquired visual observations. If the target object is not detected, the system operates in the frontiers exploration mode; otherwise, it switches to the goal navigation mode. In the exploration mode, the frontier point with the highest semantic value is selected from the semantic value map and provided as the next waypoint for the local planner.

3.2.1 Frontiers Exploration

Frontier Map Generation. A frontier map is a 2D top-down grip map constructed from depth image and odometry, which encodes the explored area, unexplored area, and frontier points [19]. The map is defined in a global reference frame aligned with the ground plane and discretized into square cells. Each cell (i, j) stores an occupancy state $o_{i,j} \in \{\text{free}, \text{occupied}, \text{unknown}\}$, as well as an explored indicator $e_{i,j} \in \{0, 1\}$, where $e_{i,j} = 1$ denotes the cell has been explored.

At each time step, the depth image is back-projected into a 3D point cloud and transformed from the camera frame to the global map frame using the agent pose, and points that are too short or too tall are discarded as non-obstacles, while the remaining points are treated as obstacles and projected onto the grid to mark the corresponding cells as occupied. Cells that are observed along the rays but contain no obstacle are marked as free.

The explored area is updated based on the agent's location and heading while accounting for occlusions from observed obstacles. Cells within agent's current field of view (FOV) are marked as explored up to the first encountered obstacle along each viewing ray. Frontier cells are then identified as the boundary between explored and unexplored areas, and the midpoints of the boundaries are used as frontier points. The quantity and location of the frontier points are changing dynamically as exploration proceeds until no frontier points remain.

Semantic Value Map Generation. The semantic value map aims to estimate the semantic relevance between spatial locations and the target object based on the agent's visual observations. Intuitively, it enables the agent to in-

fer how likely a target object may exist in regions beyond the current FOV by accumulating semantic evidence from past observations. By prioritizing regions with higher semantic relevance, the value map guides the agent toward more promising directions, thereby improving exploration efficiency. In this regard, a pre-trained VLM (e.g., BLIP2 [10], CLIP [12]) can be used to calculate the cosine similarity between the observed RGB image and a text prompt containing the target, in which a higher similarity score indicates stronger semantic relevance.

While the VLM can provide semantic information at the level of the current visual observation, the agent requires spatially grounded representations to navigate. Consequently, a transformation from image-level semantics to spatial semantics is required. To this end, the similarity score is back-projected from the image space to the top-down occupancy map using the depth image and agent pose, and assigned to each map cell within the area observed in the current FOV. As the agent explores the environment, semantic values from multiple observations are incrementally accumulated to construct a global semantic value map.

Since the semantic value map is a long-term estimation of a spatial location, a confidence score is introduced to reflect the reliability of each semantic observation for this iteratively generated map [19]. Formally, the confidence score is defined as $c = \cos^2\left(\frac{\theta}{\theta_{fov}/2} \times \frac{\pi}{2}\right)$, where θ_{fov} is the agent's horizontal FOV and θ is the angle between the optical axis.

When the agent's FOV overlaps with the previously visited regions, the semantic value of each map cell (i, j) is updated by fusing the current observation with the existing value. Specifically, the updated semantic value $v_{i,j}^{new}$ is computed as a confidence-weighted average of the current semantic value $v_{i,j}^{curr}$ and the previous value $v_{i,j}^{prev}$:

$$v_{i,j}^{new} = \frac{c_{i,j}^{curr}\, v_{i,j}^{curr} + c_{i,j}^{prev}\, v_{i,j}^{prev}}{c_{i,j}^{curr} + c_{i,j}^{prev}} \tag{1}$$

The confidence score is also updated by weighted average:

$$c_{i,j}^{new} = \frac{(c_{i,j}^{curr})^2 + (c_{i,j}^{prev})^2}{c_{i,j}^{curr} + c_{i,j}^{prev}} \tag{2}$$

Waypoint Navigation. In this study, a PointGoal Navigation (PointNav) policy, trained via distributed deep reinforcement learning algorithm [17], is adopted as the local planner to generate actions toward a specified waypoint. PointNav aims to reach a specified 2D waypoint while relying only on geometric cues from onboard sensing (i.e., visual observations and odometry), without requiring semantic understanding. This PointNav policy takes as input the egocentric depth image along with the relative goal specification, namely the distance and heading from the agent to the desired waypoint, and does not use RGB images.

3.2.2 Goal Navigation

Object Detection. Once the target object is detected, the agent switches from frontiers exploration to goal navigation by computing a goal waypoint at the nearest point on the detected object through the segmentation and depth, and then approaches this waypoint via a local planner. To be more specific, the object detector first generates a bounding box around the target. Conditioned on the RGB observation and the detected bounding box, an object segmentor predicts the contour of the target object, yielding a pixel-level object area for subsequent geometric localization. By integrating the extracted contour with the depth image and the agent's pose, the nearest 3D point on the target object surface to the agent, namely the goal waypoint, can be obtained. This point is subsequently provided to the local planner to generate a sequence of navigation actions.

4 Experiments

4.1 Experimental Setup

Implementations. For the VLM to compute the cosine similarity between the RGB images and a text prompt through vision-language representations,we adopt a pre-trained BLIP-2 [10]. The model is used solely for inference without any task-specific fine-tuning. Regarding the object detector, either YOLOv7 [16] or Grounding-DINO [11] is conditionally selected according to whether the target object belongs to pre-defined COCO categories. To balance detection reliability across seen and unseen categories, we applied different confidence thresholds, setting them to 0.8 for COCO classes and 0.4 for non-COCO classes, respectively. After obtaining the bounding box of the target object, Mobile-SAM [20] is employed as a box-prompted segmentor to extract a pixel-level object mask. Notably, all adopted models are kept frozen throughout the experiments. Key configurations for the robotic agent are summarized in Table 1.

A combination of YOLOv7/GroundingDINO and MobileSAM is adopted as the baseline perception models for target detection and segmentation. To assess the impact of stronger perception capabilities, this baseline is replaced with SAM3 [5] as a stronger, unified perception alternative. In addition, a greedy frontier-based navigation strategy that selects the closest frontier point as the next waypoint is implemented as a non-semantic exploration baseline, which relies solely on geometric information and without incorporating any vision-language guidance.

Simulator and Task Setup. The task is formulated as a single-target object navigation task, where the agent is

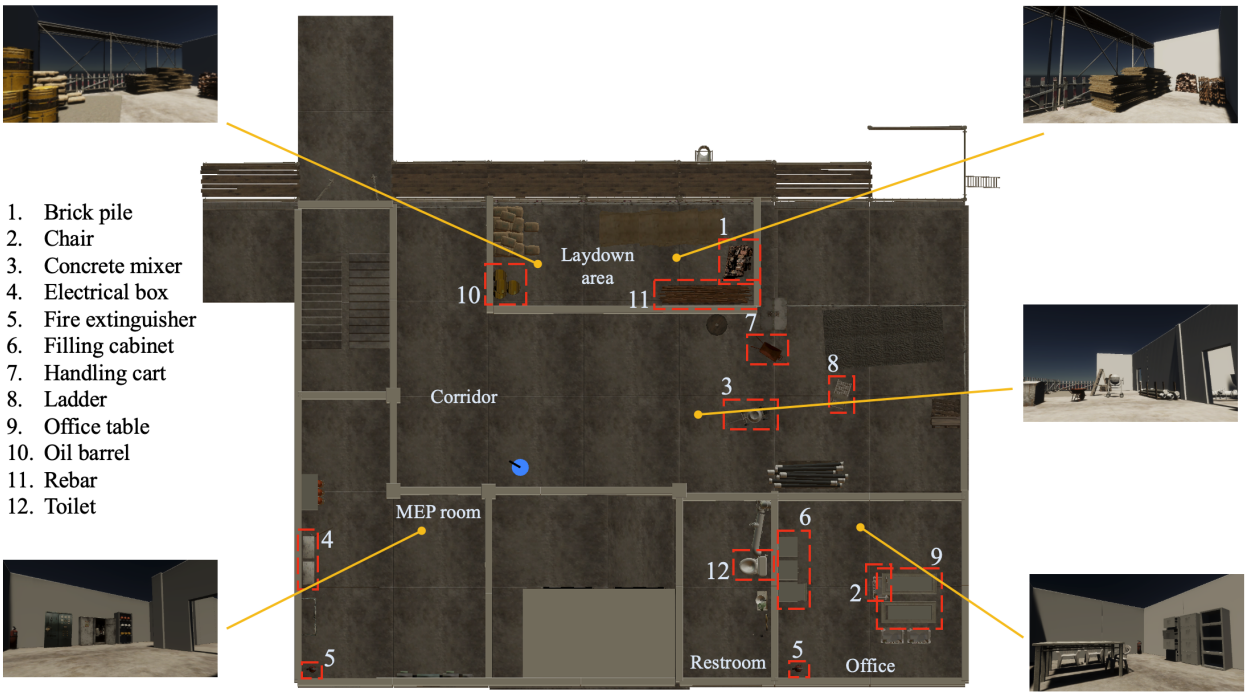

Figure 2. Top-down view of the simulated environment showing the agent (blue circle) with its initial position and orientation (black line). Target objects in this experiment are indicated by dotted lines.

Table 1. Agent Configurations

Configuration	Value
Camera height	0.88 m
Agent radius	0.18 m
Horizontal field of view	79°
RGB sensor resolution	640×480
Depth sensor resolution	640×480

required to locate one specified object per episode. All experiments are conducted using the Habitat [13] simulator. Due to the lack of standardized benchmarks for semantic navigation in construction environments, we design a custom indoor construction site in Unity (Figure 2) and integrate it into Habitat to approximate realistic scenarios. The total navigable area of the simulated environment is approximately 367 m^2 and features unstructured object layouts distributed across multiple functional areas, including corridors, MEP room, laydown area, restroom and temporary office. The environment contains a total of 26 object categories, and the target objects selected for the experiment are depicted in Figure 2. Additionally, to examine the robustness of the navigation system toward the variations in scene layout and target placement, we generated an additional construction environment using an LLM-driven layout generator [18]. The generated scene (i.e., Scene B) preserves the same functional areas as the original environment (i.e., Scene A), while varying their spatial arrangement. Objects were then re-instantiated and randomly placed within their corresponding functional areas. All experiments are performed on a single NVIDIA RTX 5090 GPU.

Evaluation Metrics. To evaluate navigational efficiency and proximity, we adopt two commonly used metrics: Success Rate (SR) and Success-weighted Path Length (SPL). SR measures the proportion of successful episodes and is defined as $\text{SR} = \frac{1}{N}\sum_{i=1}^{N} S_i$ where $S_i \in \{0, 1\}$ indicates whether episode i is successful, and N denotes the total number of episodes. SPL measures navigation efficiency by comparing the executed path length l_i to the geodesic shortest path length p_i, and is defined as $\text{SPL} = \frac{1}{N}\sum_{i=1}^{N} S_i \cdot \frac{p_i}{\max(p_i, l_i)}$

4.2 Results

Quantitative Results. The results of all episodes under the baseline navigation setting are analyzed and summarized in Table 2. Three primary failure modes corresponding to perception errors and VLM-guidance failures have been identified:

1. **False Negative (FN)**: The target object is within the agent's observation but is not recognized by the detection model. In categories such as *Electrical box* and *Rebar*, the VLM provided informative guidance

that brings the agent into viewpoints where the target is visible. However, the detection model fails to identify the target object, causing the agent to repeatedly explore nearby frontiers and eventually exhaust the step budget.

2. **False Positive (FP)**: The detection model incorrectly identifies a non-target object as the target. This failure typically arises from inter-class visual ambiguity, where objects with similar shape or color are mistaken for the target (e.g., a wooden pile misidentified as a bricks pile), leading to premature termination of the episode.
3. **Not Encountered (NE)**: The agent fails to reach any viewpoint from which the target object is observable. For objects such as *Oil barrel*, the VLM fails to provide meaningful guidance, causing the agent to repeatedly exploring previously visited areas since the newly discovered frontiers have lower semantic value than those already explored, the agent prioritizes revisiting known regions. This issue particularly severe when new frontiers are distant from previously visited areas, leading the agent to spend a large number of steps oscillating between these areas and never explores the region where the target object is located.

Table 2. Task Performance and Failure Modes by Object Category.

Object Category	Success	SPL(%)	Failure Mode
Electrical box	✗	0	FN
Fire extinguisher	✓	100	–
Toilet	✓	25	–
Office table	✗	0	FP
Chair	✗	0	FP
Filing cabinet	✗	0	NE
Ladder	✓	100	–
Bricks pile	✗	0	FP
Rebar	✗	0	FN
Oil barrel	✗	0	NE
Concrete mixer	✓	94	–
Handling cart	✗	0	FN

Comparative Analysis on Perception and Exploration Strategies. Based on the quantitative results, we perform a controlled substitution analysis to disentangle the effects of target detection models and exploration strategies, specifically the presence or absence of vision–language–based semantic guidance, on overall navigation performance. In the perception analysis, all navigation components, hyperparameters, and evaluation settings are kept unchanged, and only the perception module responsible for target detection is modified. It is worth noting that SAM3 runs in a zero-shot and fully frozen setting. This experiment is not intended as a direct model comparison, but as a controlled substitution of the perception models to isolate its contribution to overall task success in a construction environment with long-tail object categories.

Table 3 presents navigation and detection performance across different navigation paradigms in Scenes A and B. The metrics SR, SPL, FN, FP, and NE are reported as averaged over object categories. As shown in Table 3, incorporating vision–language–based semantic guidance consistently improves navigation performance compared to purely frontier-based exploration. Specifically, the success rate increases from 25% under the frontier-based strategy to 33% when semantic guidance is introduced using the baseline perception models.

Integrating SAM3 yields improvement, boosting the success rate from 33% to 58% and SPL from 26% to 38%. Detection performance is also enhanced, as evidenced by the sharp decline in FN (33% to 8%) and FP (25% to 8%) rates, demonstrating the perception module has a critical impact on overall navigation performance in complex construction environments and can benefit from more powerful detection models. Notably, episodes that previously terminated early due to incorrect detections now prolong into aimless exploration, resulting in an increase in NE (16% to 25%) when the VLM fails to provide effective guidance. This shift suggests that, beyond perception, exploration efficiency is still constrained by weak and noisy semantic cues in construction settings, which limits SPL even when detections become more reliable.

Table 3. Comparison across different perception and exploration paradigms under different scene configurations.

Scene	Paradigm	SR ↑	SPL ↑	FN ↓	FP ↓	NE ↓
Scene A	P1	25%	18%	42%	17%	17%
	P2	33%	26%	33%	25%	**16%**
	P3	**58%**	**38%**	**8%**	**8%**	25%
Scene B	P1	17%	4%	50%	25%	8%
	P2	33%	5%	25%	**8%**	**33%**
	P3	**42%**	**23%**	**17%**	**8%**	**33%**

Note: P1, P2, and P3 denote Frontier-based (without VLM), YOLO/GroundingDINO + MobileSAM, and SAM3, respectively.

A further comparison between Scene A and Scene B reveals that Scene B is consistently more challenging for navigation. Although semantic-guided paradigms still outperform purely frontier-based exploration, their gains are reduced under Scene B. In particular, SPL drops substantially for all paradigms (e.g., from 26% to 5% for P2), indicating that the main difficulty lies not only in reaching the target, but also in navigating efficiently. This suggests that randomized scene configurations lead to longer trajectories and less effective exploration, even when the target is eventually found.

Moreover, the performance degradation in Scene B is

not always accompanied by uniformly worse FN or FP rates, implying that the lower navigation performance cannot be solely attributed to poorer detection. Instead, it highlights that scene layout and target placement also influence how effectively semantic predictions can be translated into navigation decisions. While SAM3 remains the best-performing perception model in both scenes, the performance gap between SAM3 and the baseline methods narrows in Scene B, suggesting that under more variable construction environments, the bottleneck increasingly shifts from perception quality alone to the joint challenge of robust semantic reasoning and efficient exploration.

5 Discussion and Conclusions

This paper investigated the applicability of vision-language based semantic navigation in unstructured indoor construction environment. Through the experiments, we examined how high-level semantic guidance and perception quality influence navigation performance under conditions of weak semantic regularities. The results demonstrate that vision-language-based navigation is feasible in construction environments yet remains fundamentally restricted. The contributions of this work are twofold. First, we validate the effectiveness of vision–language–based ObjectNav in a simulated construction-site environment, and provide an evaluation that highlights common failure modes and performance bottlenecks. Second, through a perception-models substitution study, we analyze how advances in visual perception affect downstream navigation performance in complex construction environments. These findings provide insights into both the current limitations and future development of vision–language–based navigation methods in construction contexts.

The larger drop in SPL than SR indicates reduced exploration efficiency rather than only reduced task completion. Performance degradation cannot be explained solely by worse detection metrics

While a more powerful perception model can improve the overall navigation success rate, the navigation system still struggles to navigate in semi-structured environment efficiently. In such an environment, semantically relationships are unstructured, whereas spatial regions remain plannable and exhibit regular geometric structures. This limits the ability of the VLM to provide meaningful guidance, as evidenced by frequent NE cases and a low SPL, with the best performance reaching around 38%. Furthermore, the current method does not support multi-floor navigation, which limits its practical applicability in construction settings.

Future research directions include investigating a task-agnostic navigation approach (e.g., constructing a semantic memory map that stores language-aligned visual embeddings in a spatially grounded representation) to mitigate unreliable directional biases induced by the VLM-based semantic guidance in environments with weak semantic relationships. Deploying the method in real indoor construction sites to systematically examine the sim-to-real gap could further enhance its practical applicability.

Acknowledgments

This work was supported by The Science and Technology Development Fund, Macau SAR (File no. 0101/2024/RIB2) and University of Macau (File no. SRG2023-00006-FST). The authors would like to thank Jian Sun for his useful feedback and discussions on the earlier idea and experiment set up.

References

[1] K. Asadi, H. Ramshankar, H. Pullagurla, A. Bhandare, S. Shanbhag, P. Mehta, S. Kundu, K. Han, E. Lobaton, and T. Wu. Vision-based integrated mobile robotic system for real-time applications in construction. *Automation in Construction*, 96:470–482, 2018. doi:10.1016/j.autcon.2018.10.009.

[2] K. Asadi, H. Ramshankar, M. Noghabaei, and K. Han. Real-time image localization and registration with bim using perspective alignment for indoor monitoring of construction. *Journal of Computing in Civil Engineering*, 33(5):04019031, 2019. doi:10.1061/(ASCE)CP.1943-5487.0000847.

[3] K. Asadi, A. K. Suresh, A. Ender, S. Gotad, S. Maniyar, S. Anand, M. Noghabaei, K. Han, E. Lobaton, and T. Wu. An integrated ugv-uav system for construction site data collection. *Automation in Construction*, 112:103068, 2020. doi:10.1016/j.autcon.2019.103068.

[4] W. Cai, L. Huang, and Z. Zou. Leveraging open-vocabulary object detection in goal-oriented navigation for assessing indoor energy–intensive devices. *Journal of Computing in Civil Engineering*, 39 (5):04025060, 2025. doi:10.1061/JCCEE5.CPENG-6376.

[5] N. Carion, L. Gustafson, Y.-T. Hu, S. Debnath, R. Hu, D. Suris, C. Ryali, K. V. Alwala, H. Khedr, A. Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025. doi:10.48550/arXiv.2511.16719.

[6] C.-F. Chan, P. K.-Y. Wong, X. Guo, J. C. Cheng, J. P.-C. Chan, P.-H. Leung, and X. Tao. Context-aware vision-language model agent enriched with domain-specific ontology for construction site safety mon-

itoring. *Automation in Construction*, 177:106305, 2025. doi:10.1016/j.autcon.2025.106305.

[7] 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*, 33:4247–4258, 2020. doi:10.48550/arXiv.2007.00643.

[8] X. Chen, H. Huang, Y. Liu, J. Li, and M. Liu. Robot for automatic waste sorting on construction sites. *Automation in Construction*, 141:104387, 2022. doi:10.1016/j.autcon.2022.104387.

[9] D. Hu and V. J. Gan. Semantic navigation for automated robotic inspection and indoor environment quality monitoring. *Automation in Construction*, 170:105949, 2025. doi:10.1016/j.autcon.2024.105949.

[10] 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*, pages 19730–19742. PMLR, 2023. doi:10.48550/arXiv.2301.12597.

[11] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer, 2024. doi:10.1007/978-3-031-72970-6_3.

[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*, pages 8748–8763. PMLR, 2021. doi:10.48550/arXiv.2103.00020.

[13] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9339–9347, 2019. doi:10.1109/ICCV.2019.00943.

[14] D. Shah, B. Osiński, S. Levine, et al. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In *Conference on Robot Learning*, pages 492–504. PMLR, 2023. doi:10.48550/arXiv.2207.04429.

[15] J. Tuomisto, Y. Zheng, I. Chauhan, O. Seppänen, and J. Pajarinen. Automating on-site object inspection with a quadruped robot and bim. *Automation in Construction*, 181:106571, 2026. doi:10.1016/j.autcon.2025.106571.

[16] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7464–7475, 2023. doi:10.1109/CVPR52729.2023.00721.

[17] E. Wijmans, I. Essa, and D. Batra. Ver: Scaling on-policy rl leads to the emergence of navigation in embodied rearrangement. *Advances in Neural Information Processing Systems*, 35:7727–7740, 2022. doi:10.48550/arXiv.2210.05064.

[18] C. Xiang, R. Bao, B. Feng, W. Wu, Z. Liu, Y. Guan, and L. Liu. Co-layout: Llm-driven co-optimization for interior layout. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(17):14371–14379, Mar. 2026. doi:10.1609/aaai.v40i17.38452.

[19] 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)*, pages 42–48. IEEE, 2024. doi:10.1109/ICRA57147.2024.10610712.

[20] C. Zhang, D. Han, Y. Qiao, J. U. Kim, S.-H. Bae, S. Lee, and C. S. Hong. Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023. doi:10.48550/arXiv.2306.14289.

[21] G. Zhou, Y. Hong, and Q. Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 7641–7649, 2024. doi:10.1609/aaai.v38i7.28597.