IndustryNAV: Exploring Spatial Reasoning of Embodied Agents in Dynamic Industrial Navigation

Yifan Li[1]*, Lichi Li[6]*, Anh Dao[1]*, Xinyu Zhou[1], Yicheng Qiao[1], Zheda Mai[3], Daeun Lee[2], Zichen Chen[4], Zhen Tan[5], Mohit Bansal[2], Yu Kong[1]

[1]Michigan State University, East Lansing, MI 48824, USA. {liyifa11, anhdao, qiaoyich, zhouxi63, yukong}@msu.edu

[2]University of North Carolina at Chapel Hill, Hill Hall, Chapel, NC 27514, USA. {daeun, mbansal}@cs.unc.edu

[3]Ohio State University, Columbus, OH 43210, USA. mai.145@osu.edu

[4]University of California, Santa Barbara, Santa Barbara, CA 93106, USA. zichen_chen@ucsb.edu

[5]Arizona State University, Tempe, AZ 85281, USA. ztan36@asu.edu

[6]Independent researcher, lichili233@gmail.com

Abstract

While Visual Large Language Models (VLLMs) show great promise as embodied agents, they continue to face substantial challenges in spatial reasoning. Existing embodied benchmarks largely focus on passive, static household environments and evaluate only isolated capabilities, failing to capture holistic performance in dynamic, real-world complexity. To fill this gap, we present IndustryNav, the first dynamic industrial navigation benchmark for active spatial reasoning. IndustryNav leverages 12 manually created, high-fidelity Unity warehouse scenarios featuring dynamic objects and human movement. Our evaluation employs a PointGoal navigation pipeline that effectively combines egocentric vision with global odometry to assess holistic local-global planning. Crucially, we introduce the "collision rate" and "warning rate" metrics to measure safety-oriented behaviors and distance estimation. A comprehensive study of nine state-of-the-art VLLMs (including models such as GPT-5-mini, Claude-4.5, and Gemini-2.5) reveals that closed-source models maintain a consistent advantage; however, all agents exhibit notable deficiencies in robust path planning, collision avoidance and active exploration. This highlights a critical need for embodied research to move beyond passive perception and toward tasks that demand stable planning, active exploration, and safe behavior in dynamic, real-world environment.

1. Introduction

Humans possess visual–spatial intelligence that enables them to perceive, manipulate, and mentally represent spatial relationships among objects, as well as to navigate within complex environments. Recent visual large language models (VLLMs) have shown remarkable effectiveness as embodied agents [47] across a wide range of tasks [28], including perception [33, 62], manipulation [24], and navigation [53, 54], *etc*. However, spatial reasoning in embodied agents [10, 52], such as measuring the distance, direction, or understanding spatial relationships, remains a fundamental and persistent challenge [16].

Recent research has dedicated great efforts to uncovering the limitations of VLLMs in spatial reasoning. Most benchmarks focus on household scenarios, such as kitchens or homes, and design vision-language question-answer (QA) pairs across multiple tasks to evaluate the visual-spatial intelligence of VLLMs. However, two critical limitations

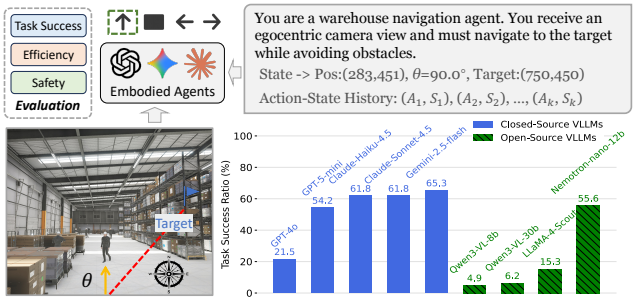


Figure 1. Illustration of IndustryNav benchmark. IndustryNav provides a zero-shot navigation setting where an embodied agent is prompted with an egocentric image, global odometry, and action–state history to reach a target while avoiding dynamic obstacles. The task-success results of nine VLLMs show that spatial reasoning in dynamic environments remains challenging, with closed-source models outperforming open-source ones; only Nemotron approaches closed-source performance.

*Equal contribution

remain. Firstly, current benchmarks mainly perform passive spatial perception within static household environments (like images [10, 14, 32, 42, 44, 57] or videos [27, 33, 52]) rather than active interaction with diverse and dynamic environments, lacking realistic dynamics such as moving objects and humans. Secondly, most benchmarks evaluate isolated reasoning skills, *e.g.*, object localization or relative position understanding, instead of holistic visual-spatial intelligence integrating perception, planning, and action. As a result, it remains unclear how well current VLLMs can perform active spatial reasoning in complex and dynamic environments.

To overcome these limitations, we present IndustryNav (see Fig. 1), an active and dynamic industrial navigation environment designed to evaluate the spatial reasoning abilities of embodied agents. Specifically, we manually create 12 dynamic warehouse scenarios using Unity, guided by 5 experts. This environment supports high-fidelity simulation of diverse warehouse assets and dynamic scenarios (moving objects and humans). To evaluate holistic spatial reasoning ability, we propose a PointGoal navigation pipeline that provides both egocentric images and global odometry information. Such a pipeline is effective in fully leveraging the global and local planning capabilities of VLLMs to complete the navigation task. Furthermore, to comprehensively evaluate the proficiency of VLLMs, we introduce five metrics across three key dimensions, *i.e.*, task success, trajectory efficiency, and decision-making safety. These metrics provide a fine-grained assessment of spatial awareness and safety-oriented reasoning, complementing traditional navigation success measures and offering deeper insights into VLLM performance in dynamic environments. Our contributions are threefold:

- We propose the *first* dynamic industrial navigation benchmark built on Unity to evaluate and advance the spatial reasoning abilities of embodied agents, allowing active interactions with dynamic environments. To support this benchmark, we manually construct a series of realistic industrial simulation scenarios to facilitate future research. Furthermore, we conduct a comprehensive evaluation of state-of-the-art VLLMs, including five closed-source and four open-source models, and derive several key insights.
- We present an effective zero-shot navigation pipeline designed to evaluate the local-global planning abilities of current VLLMs. The pipeline decomposes the PointGoal navigation into local-global planning and action stages, allowing VLLMs to jointly analyze egocentric observations and global odometry information to generate coherent and goal-directed actions.
- We introduce two new metrics "collision rate" and "warning rate" to better evaluate the agent's ability to estimate spatial distance and maintain safe navigation behaviors. The collision rate quantifies the proportion of episodes where the agent physically collides with surrounding objects or humans. In contrast, the warning rate measures the frequency at which an agent approaches a safety-critical distance threshold, such as within a predefined buffer zone around obstacles.

Table 1. Comparison with existing spatial reasoning benchmarks. "Colli." represents whether collision evaluation is supported; "Dyn." denotes the presence of moving objects; "L–G" reflects local–global planning ability evaluation; and "Active" indicates whether active interactions with environments are supported.

Benchmark	Scenario	Platform	Colli.	Dyn.	L-G	Active
SpatialRGBT-Bench [10]	Household	Omni3D	✗	✗	✗	✗
EmbSpatial-Bench [15]	Household	AI2-THOR, Habitat	✗	✗	✗	✗
RoboSpatial [42]	Household	Mixed Indoor sim	✗	✗	✗	✗
VSI-Bench [52]	Household	ScanNet/++, ARKitScenes	✗	✗	✗	✗
Q-Spatial [30]	Household	ScanNet	✗	✗	✗	✗
3DSRBench [32]	In-the-wild	COCO	✗	✗	✗	✗
Open3D-VQA [57]	Outdoor	UrbanScene3D, EmbodiedCity	✗	✗	✗	✗
CA-VQA [14]	Household	CA-1M, ARKitScenes	✗	✗	✗	✗
OpenEQA [33]	Household	ScanNet, HM3D	✗	✗	✗	✓
VSP [46]	Maze Game	OpenAI-Gym	✗	✗	✗	✓
BEHAVIOR-1K [26]	Household	OmniGibson	✗	✗	✗	✓
LoTa-Bench [12]	Household	AI2-THOR, VirtualHome	✗	✗	✓	✓
PARTNR [7]	Household	Habitat 3.0	✗	✓	✓	✓
IndustryNav	Warehouse	Unity	✓	✓	✓	✓

2. Related Work

This section reviews studies on spatial reasoning, including VLLMs and benchmarks, as well as agent navigation.

Spatial Reasoning of VLLMs. Recent VLLM research has explored several strategies to improve spatial reasoning, which can be grouped by the spatial representations they use. *Explicit geometric pretraining* approaches learn spatial knowledge from large-scale 3D data (3D-VLA [60]) and annotated scene datasets (SpatialVLM [9], SpatialLLM [25]), enabling models to understand geometric relationships and spatial concepts (SpatialThinker [57]). *Depth-aware perception* approaches strengthen 2D understanding by adding geometric cues such as depth maps [10, 15], point clouds [50], and multi-modal feature projection [19, 49]. *Structured relational reasoning* methods build symbolic or graph-based representations [6], using scene graphs for relational reasoning [45], topological maps for global planning [54], and hierarchical compositions for concept-level understanding [30]. These representations offer more interpretable spatial reasoning.

Spatial Reasoning Benchmarks. As vision-language models gain stronger spatial intelligence, evaluation has progressed from simple geometric reasoning [23] to complex embodied interactions (ScanQA [3]). Passive perception benchmarks assess spatial understanding from static views, focusing on visual grounding [22, 42], spatial relationship inference (Q-Spatial [30]), and multi-view scene layout understanding [10, 15] without any interaction with the environment [32, 58]. Active exploration benchmarks add multi-step navigation, incorporating strategic search

and question-driven exploration (OpenEQA [33]), generalization to unseen scenes [12, 26]), and integrate reasoning for embodied QA [13], emphasizing exploration efficiency under simplified dynamics [7]. More recent efforts address domain-specific needs, such as outdoor navigation [20], urban spatial understanding [57], and augmented reality tasks focusing on spatial anchoring and cross-reality alignment [14]. These works highlight that spatial reasoning requirements vary widely across applications. As shown in Tab. 1, IndustryNav provides a comprehensive evaluation covering collision awareness, dynamic obstacle handling, local–global planning, and active interaction, supported by quantitative safety metrics essential for industrial applications, whereas prior benchmarks address only subsets of these aspects [12, 26].

Agent Navigation. Agent navigation aims to equip embodied models with the ability to perceive, reason, and act in complex environments [61]. Prior work can be grouped by the reasoning paradigm it follows. Earlier research focused on instruction-following reasoning [4], emphasizing language grounding for command understanding (VLN [1]), action alignment for trajectory generation [38], and semantic mapping for environment comprehension [31, 41]. Subsequent research has concentrated on exploration-based reasoning. These studies focus on three key approaches: optimizing for information gain (*e.g.*, EQA [13]), employing efficient search strategies [11, 59], and adapting exploration policies to enhance overall task performance [29, 56]. More recent advances introduce structured reasoning paradigms that utilize hierarchical planning for long-horizon tasks (SG-Nav [54]), topological reasoning for global navigation [18], and semantic graphs for open-vocabulary localization [21], supporting interpretable and compositional spatial reasoning [55]. Building upon these advances, our work extends agent navigation toward spatial reasoning in dynamic industrial environments and proactive interaction for reliable navigation.

3. IndustryNav

This section provides comprehensive details on the IndustryNav environment, specifically describing the creation of industrial scenarios, the implemented navigation pipeline, and our evaluation metrics.

3.1. Industrial Scenario Creation

This subsection introduces the industrial scenario, including the simulation environment and its design details.

Simulation Environment. Our industrial scenarios are built using Unity, a robust cross-platform and real-time 3D development engine most commonly known for video game applications. We choose Unity as our simulator for three reasons. Firstly, it provides a large-scale, high-fidelity asset necessary for various functions for constructing realistic warehouses and benefits from strong community support. Secondly, Unity is highly optimized for various hardware and ensures multi-platform compatibility (Mac, Linux, Windows), while allowing for headless execution. Thirdly, it integrates seamlessly with the open-source ML-Agents toolkit, officially offered by Unity, which facilitates the training of embodied agents through reinforcement and imitation learning with visual feed support.

Warehouse Creation. To construct dynamic industrial scenarios in Unity, we design two types of scenarios, *i.e.*, dynamic and static, distinguished by whether they involve objects or human animations. A team of five experts creates 12 dynamic warehouse scenes by manually populating large empty spaces with diverse assets. The layouts are designed to depict realistic warehouse configurations that can be found online. Leveraging an active community, we utilize a broad collection of 3D warehouse assets covering architecture (walls, floor markings, lighting, stairs, supporting beams), storage systems (racks, barrels, containers, conveyor belts, boxes), material handling equipment (forklifts, hand trucks, reachlifts, ground cargo robots), and personnel and amenities (workers, chairs, safety signage, trash bins, fire extinguishers). The various warehouses also intentionally encompass a variety of more and less safe scenarios, *e.g.*, perfectly bestowed versus poorly angled containers, closed versus opened cargo containers, inflammable liquid barrels, improper object occupation or obstruction of common pathways to mimic realistic cargo transitory states during processing and handling. These are inspired by OSHA safety regulations defined by the United States government. All scenes and assets are rendered under the High-definition rendering pipeline (HDRP) to retain high graphical fidelity with careful setup on material reflections, global and local illumination, and various shadow rendering adjustments. To improve runtime frame rates, we even employ level-of-detail (LOD) 3D model dynamic switching based on the camera rendering distance to certain art assets.

Unlike most existing spatial reasoning benchmarks (see Tab. 1), IndustryNav includes dynamic objects (like forklifts or robots) and workers. To enable these dynamic behaviors, we manually design the trajectories of selected objects and workers using the Splines package in Unity. Each trajectory is carefully crafted to reflect realistic motion patterns observed in industrial environments. These dynamics introduce temporal and spatial variability, making IndustryNav a more realistic and challenging benchmark for evaluating the spatial reasoning and navigation abilities of embodied agents. Furthermore, to detect collisions between the agent and surrounding obstacles, we assign hand-crafted collider geometries to all objects in the scene, thereby resolving potential clipping issues. This setup enables precise collision detection and response within Unity's physics engine, allowing agents to perceive and react to obstacles such

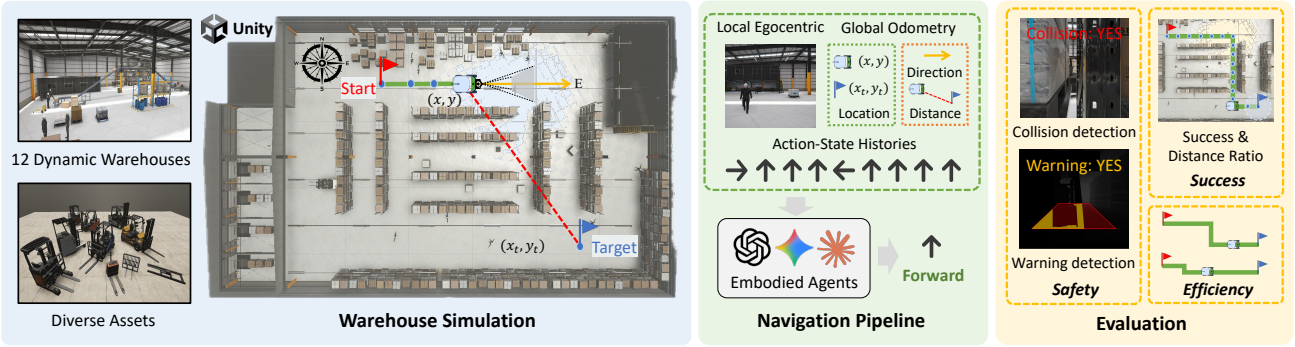


Figure 2. Overview of the IndustryNav benchmark. IndustryNav is built on Unity and consists of 12 dynamic warehouse environments. The navigation pipeline combines local egocentric observations with global odometry information, enabling the embodied agent to generate appropriate navigation actions. To comprehensively evaluate navigation performance, we assess three dimensions: task success (Success Ratio and Distance Ratio), trajectory efficiency (Average Steps), and safety behaviors (Collision Ratio and Warning Ratio).

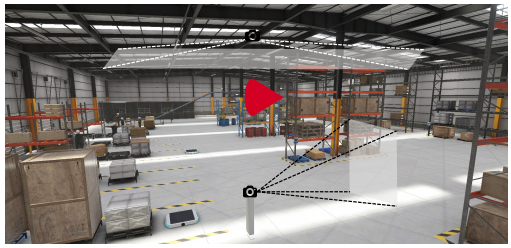


Illustration of agent cameras

Egocentric view Top-down view

Figure 3. Visualization of the camera setup for the IndustryNav agent. An egocentric camera mounted on the agent's body captures first-person visual observations of the surrounding environment, while a fixed bird's-eye-view camera positioned above the warehouse continuously tracks the agent's movement in real time using a red cone marker. The bottom panels show the corresponding egocentric and top-down visual perspectives used for navigation monitoring and trajectory analysis.

as walls, shelves, forklifts, and human workers.

To capture both local and global information from the agent, we employ a multi-sensor tracking setup shown in Fig. 3. An egocentric camera with a resolution of 1024×1024 is attached to the agent's body, providing localized visual observations of its immediate surroundings. Simultaneously, the agent tracks its global state by recording its current coordinate (x, y) and orientation angle $\theta \in [0, 360]$ after every step. For real-time visual monitoring, a small red cone is placed on the top of the agent's head, which is tracked by a fixed bird's-eye-view camera (resolution: 1024×512) mounted high above the warehouse. This top-down view provides a visual reference to monitor the agent's pixel location and progress toward the target during navigation. This combination of egocentric vision and global top-down tracking ensures both accurate trajectory visualization and robust quantitative analysis of the agent's navigation behaviors within the warehouse environment.

3.2. Navigation Pipeline

To comprehensively evaluate the spatial reasoning ability (local-global planning) of embodied agents in dynamic industrial scenarios, we design an effective navigation pipeline under the zero-shot setting. Inspired by PointGoal navigation [8, 37, 39, 40, 43, 48], which trains a navigation model to reach a specified target location using egocentric visual inputs and relative goal information, we extend this paradigm to evaluate zero-shot VLLMs rather than task-specific navigation models. In our navigation pipeline, we explicitly provide the agent's current state information obtained from the simulator, *i.e.*, its location, direction, distance, egocentric image, and action-state histories, and allow the embodied agent to generate the final action.

Specifically, as shown in Fig. 2, our navigation pipeline integrates local egocentric image, global odometry information, and action-state histories to guide the embodied agent's decision-making. The egocentric image enables the agent to perceive its immediate surroundings and avoid potential collisions. By continuously analyzing these images, the agent can detect nearby obstacles, dynamic objects, and environmental changes in real time, allowing it to make informed decisions for safe and efficient navigation.

The global odometry, which includes the agent's current and target locations, direction, and distance towards the target, acts as a spatial reference that guides the navigation process. To help the agent better understand the relationship between its current orientation θ and the environment coordinate system, we explicitly provide the following mapping as part of the prompt:

$\theta = 0^\circ \rightarrow$ West, $90^\circ \rightarrow$ North, $180^\circ \rightarrow$ East, $270^\circ \rightarrow$ South

This mapping enables the agent to align its internal directional reasoning with the real-world spatial coordinate system, facilitating more accurate navigation decisions. By continuously updating this information after each step, the agent can maintain an accurate understanding of the global position and plan more efficient paths toward the goal. Additionally, the action-state histories (see the t-step history below) are provided to offer temporal context about the past behaviors towards the target, helping the agent identify repetitive failures and avoid getting stuck in action loops.

Step t: Position $(100, 200)$, $\theta = 90^\circ$, Action: `turn left`, Distance to target: 30, Target (130, 200)

After perceiving the local and global information, the agent should select an appropriate action from a discrete action space consisting of `forward`, `turn left`, `turn right`, and `stop`. We define the corresponding action signals for these commands to control the agent within the environment based on the ML-Agents library. It is worth noting that, for simplicity, all turning actions involve a fixed rotation of 90 degrees. The agent should generate `stop` if the agent reaches near the target. We also provide the agent with an action-state mapping prompt to help it learn the correspondence between actions and positional changes.

Action–State Mapping

Turning (changes θ only, position unchanged)
- `turn_right`: $\theta = (\theta + 90) \bmod 360$
- `turn_left`: $\theta = (\theta - 90) \bmod 360$

Moving (changes position only, θ unchanged)
- When $\theta = 0^\circ$ (West): `forward` $\rightarrow (-34,\ 0)$
- When $\theta = 90^\circ$ (North): `forward` $\rightarrow (0,\ -34)$
- When $\theta = 180^\circ$ (East): `forward` $\rightarrow (+34,\ 0)$
- When $\theta = 270^\circ$ (South): `forward` $\rightarrow (0,\ +34)$

To analyze the underlying rationale of the final action, we also require the agent to output the reasoning process (see Fig. 5) for that decision. This concurrent output allows for a detailed evaluation of its decision-making integrity and a deeper understanding of its behavioral strategy.

This navigation framework enables the agent to continuously perceive its surroundings, reason about spatial relationships, and adjust its actions. By leveraging both local and global information, the agent maintains situational awareness, allowing it to adapt its trajectory to dynamic environmental changes. Consequently, the agent is expected to effectively avoid obstacles and efficiently navigate toward the target point within dynamic industrial scenarios.

3.3. Evaluation

We propose five metrics to comprehensively evaluate the agent's performance across three key dimensions, *i.e.*, task success (Success Ratio and Distance Ratio), trajectory efficiency (Average Steps), and decision-making safety (Collision Ratio and Warning Ratio).

Success Ratio (SR). Let d_i denote the distance between the agent's position and the target at the final step T during the i-th run. The SR metric is then defined as:

$$\mathrm{SR} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}[d_i \leq \delta], \tag{1}$$

where N is the total number of runs across different scenes, $\mathbb{1}[\cdot]$ is the indicator function that equals 1 if the condition holds and 0 otherwise, and δ is a predefined distance threshold, which we set to 20 in our experiments. A higher SR value indicates that the agent consistently maintains a smaller distance to the target across different runs, reflecting both stable control and reliable navigation performance.

Distance Ratio (DR). While SR captures the overall success rate of each model, it is too coarse to reflect fine-grained performance, particularly for failed runs. To solve this, we introduce the DR metric to quantify the agent's relative progress toward the target across all runs. Let D_i denote the initial distance between the starting point and the target in the i-th run, and d_i denote the final distance after navigation. Then the DR metric is given by:

$$\mathrm{DR} = \frac{1}{N} \sum_{i=1}^{N} \frac{D_i - d_i}{D_i}. \tag{2}$$

A higher DR value indicates that the agent is able to reduce a larger proportion of the initial distance to the target, thereby achieving more effective navigation behavior, even when the target is not fully reached.

Average Steps (AS). To evaluate the efficiency of the trajectories planned by the agent, we use the AS metric, which measures the mean number of steps taken per run:

$$\mathrm{AS} = \frac{1}{N} \sum_{i=1}^{N} T_i, \tag{3}$$

where T_i denotes the total number of steps taken in the i-th run. A lower AS value indicates that the agent reaches the target with fewer steps, reflecting a more efficient and intelligent navigation strategy in its planning process.

Collision Ratio (CR). Safety is also a crucial aspect of navigation, such as avoiding collisions or receiving warnings, beyond merely completing the task successfully. However, existing spatial reasoning benchmarks lack dedicated metrics for detecting collisions during navigation. To address this limitation, we introduce the CR metric, which

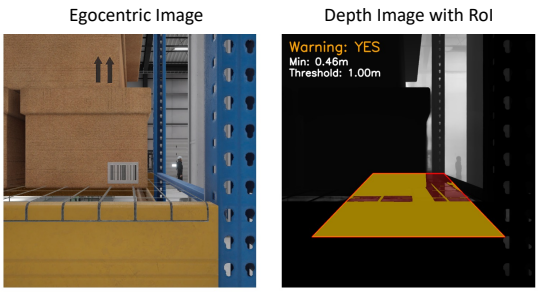


Figure 4. An illustration of the warning detection. The warning is triggered when the minimum depth values within the RoI region fall below a predefined threshold.

measures the proportion of collisions per run. By leveraging the colliders of all objects in Unity, collisions can be efficiently detected by verifying whether the agent's position changes as expected after executing the `forward` action. The CR is defined by:

$$\mathrm{CR} = \frac{1}{N}\sum_{i=1}^{N}\frac{C_i}{F_i}, \tag{4}$$

where C_i denotes the number of collisions, identified by checking whether the agent's position changes after executing a `forward` action, and F_i represents the total number of `forward` actions in the i-th run. A lower CR value indicates safer navigation, demonstrating the agent's ability to effectively avoid obstacles. This metric provides an intuitive measure of the agent's spatial awareness and collision-avoidance capability, complementing the SR, DR, and AS metrics that focus on goal-reaching efficiency. In dynamic industrial environments, where moving objects and workers frequently alter the navigable space, a low CR reflects the agent's capacity to adapt its trajectory in real time, ensuring safe and reliable operation.

Warning Ratio (WR). CR can detect whether collisions occur, but it does not reflect potential risks that occur before collisions happen. To address this, we propose the WR metric, which quantifies near-collision risks based on depth information. Specifically, we employ the Depth Pro method [5] to estimate depth maps from egocentric images, where each pixel value represents the distance to the nearest obstacle. As shown in Fig. 4, we then define a region of interest (ROI) along the agent's forward path to detect potential hazards. If any pixel within the ROI has a depth value smaller than a predefined threshold, the frame is marked as a warning state. The WR is defined by:

$$\mathrm{WR} = \frac{1}{N}\sum_{i=1}^{N}\frac{W_i}{T_i}, \tag{5}$$

where W_i indicates the number of warning states in the i-th run. A lower WR value implies that the agent can maintain a safer distance from surrounding objects, reflecting stronger spatial perception and proactive risk avoidance. When combined with the CR metric, WR offers a more comprehensive evaluation of the agent's safety behavior, capturing both actual collisions and potential near-miss events that occur during navigation in dynamic industrial environments.

4. Experiment

4.1. Embodied Agent Comparison

Settings. To comprehensively evaluate spatial reasoning performance on the IndustryNav benchmark, we compare nine state-of-the-art VLLMs, including five closed-source models (GPT-4o, GPT-5-mini, Gemini-2.5-flash, Claude-Haiku-4.5, and Claude-Sonnet-4.5) and four open-source models (Nemotron-nano-12B-v2-VL, Llama-4-Scout, Qwen3-VL-30B-a3B-Instruct, and Qwen3-VL-8B-Instruct), all accessed via the OpenRouter API. We evaluate only efficient VLLMs, excluding larger reasoning-oriented models, to better match the computational and latency requirements of real-world applications. All experiments are conducted on personal computers running either macOS or Windows operating systems.

We randomly sample four start–target point pairs of varying difficulty for each scenario, with each pair representing a single run for every agent. Each run consists of 70 steps. At each step, the agent is prompted with both the egocentric image and global odometry information, and it outputs the corresponding action and reasoning process in JSON format. We use five metrics, as defined in Sec. 3.3, to evaluate the model's performance. We set $\delta = 20$ as the success threshold and define 1 meter as the threshold for detecting warnings. The action–state history length is fixed to 10 across all experiments.

Results Analysis. We conduct experiments on the IndustryNav benchmark, comparing various VLLMs across three dimensions: task success, efficiency, and safety, as shown in Tab. 1. We can draw some key insights below:

❶ No VLLMs can reliably navigate to the target. Across all nine models, the Task Success Ratio remains low (less than 70%), and none of them can consistently reach the target across all scenarios. This highlights that:

> *Spatial reasoning and long-horizon navigation in dynamic industrial environments remain fundamentally challenging for current VLLMs.*

❷ Closed-source VLLMs consistently outperform open-source VLLMs. This trend is evident across the first two dimensions. For task success, closed-source models achieve substantially higher success rates than open-source ones. Most of open-source VLLMs, particularly Qwen3-VL and LLaMA-4, remain less competitive for high-risk,

Table 2. Comparison of spatial reasoning performance across nine VLLMs (five closed-source and four open-source) on the IndustryNav benchmark. All evaluated VLLMs exhibit low success rates in navigating to the target across all scenarios. Closed-source models consistently outperform open-source counterparts across most metrics, indicating stronger spatial reasoning capabilities and safer navigation behaviors in dynamic industrial environments.

Embodied Agents	**Task Success (%)**		**Efficiency**	**Safety (%)**	
	Success Ratio ↑	Distance Ratio ↑	Average Steps ↓	Collision Ratio ↓	Warning Ratio ↓
Closed-Source VLLMs					
GPT-4o [36]	21.53	49.41	66.76	7.86	13.45
GPT-5-mini	54.17	81.9	49.91	16.89	24.13
Claude-Haiku-4.5 [2]	61.81	82.87	46.80	32.18	31.57
Claude-Sonnet-4.5 [2]	61.81	86.26	47.33	27.68	31.52
Gemini-2.5-flash [17]	65.28	84.49	45.95	32.14	37.16
Open-Source VLLMs					
Qwen3-VL-8b-Instruct [51]	4.86	27.05	67.22	27.82	25.70
Qwen3-VL-30b-A3B-Instruct [51]	6.25	26.2	66.70	18.97	26.28
LLaMA-4-Scout [34]	15.28	56.4	61.53	24.38	35.06
Nemotron-nano-12b-v2-VL [35]	55.56	80.48	50.69	31.73	36.54

complex, and dynamic navigation tasks. From the efficiency perspective, closed-source models generally require fewer steps to reach the target, reflecting superior planning and route optimization abilities. This indicates that:

> *Closed-source VLLMs demonstrate significantly stronger spatial reasoning than open-source ones.*

❸ **Nemotron-nano-12B-v2-VL stands out among open-source VLLMs.** With a 55.56% Success Ratio, Nemotron's performance is approaching that of closed-source counterparts. Furthermore, it demonstrates reasonable efficiency and moderate safety scores, performing comparably in these key metrics as well. This indicates:

> *Nemotron stands out as the most promising open-source baseline for spatial reasoning.*

❹ **Safety remains a major challenge in dynamic industrial navigation.** Both closed-source and open-source VLLMs show high Collision and Warning Ratios, highlighting substantial deficiencies in hazard perception, navigating around dynamic obstacles, and maintaining consistent collision avoidance. It reveals that:

> *All VLLMs remain far from achieving the level of safety required for real-world deployment.*

4.2. Case Analysis

To gain deeper insight into the local and global planning abilities of current VLLMs, we analyze representative cases from GPT-5-mini in Fig. 5. These examples illustrate how the model arrives at both correct and incorrect navigation actions. More cases are presented in the Appendix.

The first case in Fig. 5 illustrates that GPT-5-mini exhibits a degree of spatial reasoning. The agent correctly recognizes that its path is blocked by a worker and a forklift, making `forward` movement unsafe. It then performs multi-step reasoning, integrating its risk assessment with its knowledge of the target's relative location (southeast) to formulate a final decision based on local and global information. This behavior demonstrates early signs of GPT-5-mini's ability to integrate local–global navigation planning in dynamic embodied environments.

The second case in Fig. 5 shows key failures in GPT-5-mini's global path planning and active exploration. Although the agent correctly identifies that the target is to its east, it fails to recognize that a shelf blocks the direct path and therefore does not replan a new route. It also overlooks the repeated action–state pattern, getting stuck in a loop instead of exploring alternatives. These issues reveal the model's limited ability to handle spatial constraints, escape local minima, and perform reliable global planning.

The third case in Fig. 5 illustrates the deficiency of VLLMs in capturing precise distance for safe behaviors. The agent's reasoning process indicates a misperception—it believes the center path is clear, which leads to the incorrect `forward` action and subsequent collision with the shelf. This type of perceptual error directly explains why all VLLMs perform poorly in relation to our efficiency metric (AS). These case analyses lead to the following insight:

> *Current VLLMs still struggle with robust global path planning, active exploration, and precise distance estimation in dynamic scenarios.*

4.3. Ablation Study

To evaluate the effectiveness of action-state histories and the necessity of the top-down view map, we present the ablation studies on three closed-source VLLMs, *i.e.*, Claude-Sonnet-4.5, Gemini-2.5-flash, and GPT-5-mini. Additional results are provided in the Appendix.

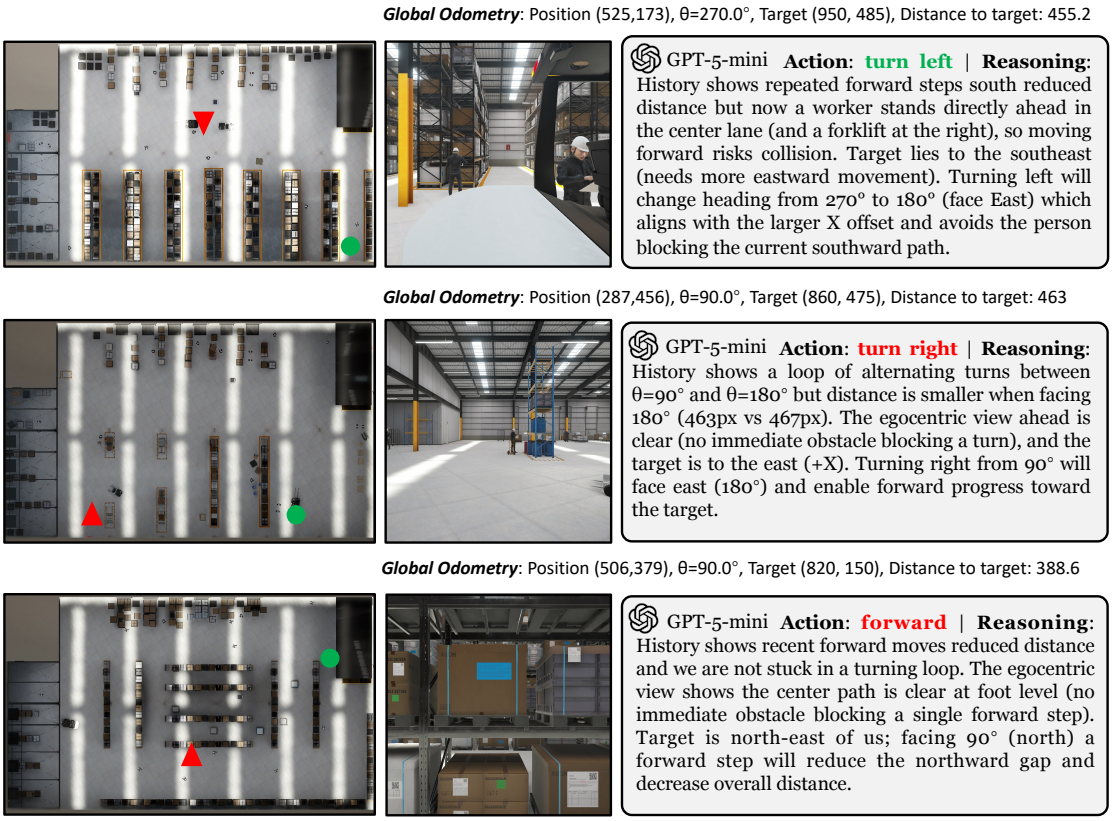


Figure 5. Illustration of both correct (first row) and incorrect (second and third rows) action behaviors of GPT-5-mini under the IndustryNav scenario. The red triangle indicates the agent's current position and direction, and the green circle marks the target location.

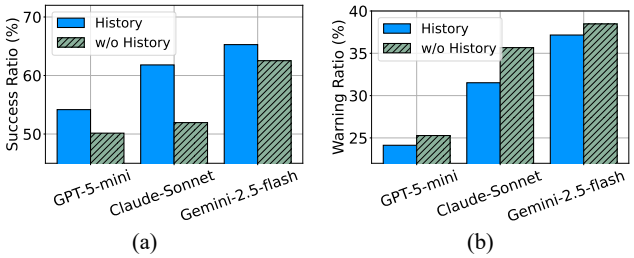


Figure 6. The effectiveness of action–state histories on (a) success ratio and (b) warning ratio for three closed-source VLLMs.

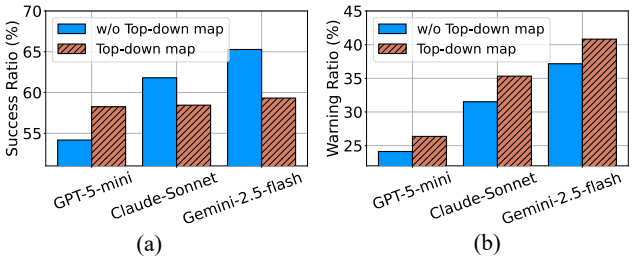


Figure 7. The necessity of a top-down view map on (a) success ratio and (b) warning ratio for three closed-source VLLMs.

Effectiveness of Action-State Histories. We remove action-state histories to assess their impact in Fig. 6. These histories provide temporal context that helps the agent detect repeated failures, leading to improved navigation success. Without this, the agent is more prone to making shortsighted decisions and repeating ineffective actions, leading to lower success and less stable trajectories. Results in Fig. 6 demonstrate its effectiveness in increasing the success ratio and avoiding collisions.

Necessity of the Top-Down View Map. Although global odometry provides precise information about the target's relative position, it lacks obstacle layout information, which is essential for global planning. To assess whether additional spatial context is beneficial, we augment the pipeline with a top-down view map, as shown in Fig. 7. The results indicate that incorporating the top-down map does not improve task success or reduce warning ratios, except for a modest gain in GPT-5-mini's success rate. We attribute this to VLLMs' limited ability to interpret top-down maps and the additional noise such representations introduce compared to clean odometry signals. Thus, we rely solely on global odometry to provide global information.

5. Conclusion

In this paper, we introduce the first benchmark of 12 manually designed Unity-based warehouses for evaluating active spatial reasoning in embodied agents. We also present a zero-shot navigation pipeline and five metrics covering task success, efficiency, and safety for comprehensive evaluation. Experiments with nine VLLMs demonstrate that spatial reasoning in dynamic industrial environments remains challenging, with closed-source models consistently outperforming open-source models, and safety remains a major limitation. Case analyses reveal that current VLLMs struggle with global path planning, active exploration, and precise distance estimation. Ablations show that action-state histories are crucial, while the top-down view map is not strictly necessary. We hope this work offers useful insights for the embodied AI community.

References

[1] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, pages 3674–3683, 2018. 3

[2] Anthropic. Claude sonnet 4.5, 2025. 7

[3] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *CVPR*, pages 19129–19139, 2022. 2

[4] Christian Bizer and Andreas Schultz. The r2r framework: Publishing and discovering mappings on the web. *COLD*, 665:97–108, 2010. 3

[5] Alexey Bochkovskiy, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. In *ICLR*, 2025. 6

[6] Wenxiao Cai, Iaroslav Ponomarenko, Jianhao Yuan, Xiaoqi Li, Wankou Yang, Hao Dong, and Bo Zhao. Spatialbot: Precise spatial understanding with vision language models. In *ICRA*, pages 9490–9498. IEEE, 2025. 2

[7] Matthew Chang, Gunjan Chhablani, Alexander Clegg, Mikael Dallaire Cote, Ruta Desai, Michal Hlavac, Vladimir Karashchuk, Jacob Krantz, Roozbeh Mottaghi, Priyam Parashar, et al. Partnr: A benchmark for planning and reasoning in embodied multi-agent tasks. In *ICLR*, 2025. 2, 3

[8] Devendra Singh Chaplot, Ruslan Salakhutdinov, Abhinav Gupta, and Saurabh Gupta. Neural topological slam for visual navigation. In *CVPR*, pages 12875–12884, 2020. 4

[9] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *CVPR*, pages 14455–14465, 2024. 2

[10] An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatialrgpt: Grounded spatial reasoning in vision-language models. In *NeurIPS*, pages 135062–135093, 2024. 1, 2

[11] Kai Cheng, Zhengyuan Li, Xingpeng Sun, Byung-Cheol Min, Amrit Singh Bedi, and Aniket Bera. Efficienteqa: An efficient approach for open vocabulary embodied question answering. *arXiv preprint arXiv:2410.20263*, 2024. 3

[12] Jaewoo Choi, Youngwoo Yoon, Hyobin Ong, Jaehong Kim, and Minsu Jang. Lota-bench: Benchmarking language-oriented task planners for embodied agents. In *ICLR*, 2024. 2, 3

[13] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *CVPR*, pages 1–10, 2018. 3

[14] Erik Daxberger, Nina Wenzel, David Griffiths, Haiming Gang, Justin Lazarow, Gefen Kohavi, Kai Kang, Marcin Eichner, Yinfei Yang, Afshin Dehghan, et al. Mm-spatial: Exploring 3d spatial understanding in multimodal llms. In *ICCV*, pages 7395–7408, 2025. 2, 3

[15] Mengfei Du, Binhao Wu, Zejun Li, Xuanjing Huang, and Zhongyu Wei. EmbSpatial-bench: Benchmarking spatial un-

derstanding for embodied tasks with large vision-language models. In *ACL*, pages 346–355, 2024. 2

[16] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. In *ECCV*, pages 148–166, 2024. 1

[17] Google DeepMind. Gemini 2.5 Flash Preview Model Card. Technical report, Google, Mountain View, CA, 2025. 7

[18] Christian Grévisse, Claude Braun, and José Batista da Costa. Autotag & tagmap: Llm-powered moodle plugins for pedagogical alignment checks. *SN Computer Science*, 6(7):1–11, 2025. 3

[19] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36:20482–20494, 2023. 2

[20] Zhenghang Hou, Weiping He, and Shuxia Wang. Enhancing visualization and interaction of complex spatial data through augmented reality. *The International Journal of Advanced Manufacturing Technology*, 134(11):5891–5906, 2024. 3

[21] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *ICRA*, pages 10608–10615. IEEE, 2023. 3

[22] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, pages 6700–6709, 2019. 2

[23] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, pages 2901–2910, 2017. 2

[24] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. 1

[25] Bo Li, Kaichen Zhang, Hao Zhang, Dong Guo, Renrui Zhang, Feng Li, Yuanhan Zhang, Ziwei Liu, and Chunyuan Li. Llava-next: Stronger llms supercharge multimodal capabilities in the wild, 2024. 2

[26] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *CoRL*, pages 80–93. PMLR, 2023. 2, 3

[27] Yifan Li, Yuhang Chen, Anh Dao, Lichi Li, Zhongyi Cai, Zhen Tan, Tianlong Chen, and Yu Kong. Industryeqa: Pushing the frontiers of embodied question answering in industrial scenarios. *arXiv preprint arXiv:2505.20640*, 2025. 2

[28] Yifan Li, Zhixin Lai, Wentao Bao, Zhen Tan, Anh Dao, Kewei Sui, Jiayi Shen, Dong Liu, Huan Liu, and Yu Kong. Visual large language models for generalized and specialized applications. *arXiv preprint arXiv:2501.02765*, 2025. 1

[29] Mingfu Liang, Ying Wu, et al. Toa: Task-oriented active vqa. *Advances in Neural Information Processing Systems*, 36:54061–54074, 2023. 3

[30] Yuan-Hong Liao, Rafid Mahmood, Sanja Fidler, and David Acuna. Reasoning paths with reference objects elicit quantitative spatial reasoning in large vision-language models. In *EMNLP*, pages 17028–17047, 2024. 2

[31] Bingqian Lin, Yunshuang Nie, Ziming Wei, Jiaqi Chen, Shikui Ma, Jianhua Han, Hang Xu, Xiaojun Chang, and Xiaodan Liang. Navcot: Boosting llm-based vision-and-language navigation via learning disentangled reasoning. *IEEE TPAMI*, 2025. 3

[32] Wufei Ma, Haoyu Chen, Guofeng Zhang, Yu-Cheng Chou, Jieneng Chen, Celso de Melo, and Alan Yuille. 3dsrbench: A comprehensive 3d spatial reasoning benchmark. In *ICCV*, pages 6924–6934, 2025. 2

[33] Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, et al. Openeqa: Embodied question answering in the era of foundation models. In *CVPR*, pages 16488–16498, 2024. 1, 2, 3

[34] Meta AI. Llama 4 model card, 2025. 7

[35] NVIDIA Nemotron Nano. Efficient hybrid mamba-transformer reasoning model. *arXiv preprint arXiv:2508.14444*, 2025. 7

[36] OpenAI. Gpt-4o system card, 2024. 7

[37] Ruslan Partsey, Erik Wijmans, Naoki Yokoyama, Oles Dobosevych, Dhruv Batra, and Oleksandr Maksymets. Is mapping necessary for realistic pointgoal navigation? In *CVPR*, pages 17232–17241, 2022. 4

[38] Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. Reverie: Remote embodied visual referring expression in real indoor environments. In *CVPR*, pages 9982–9991, 2020. 3

[39] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *CVPR*, pages 9339–9347, 2019. 4

[40] Alexander Sax, Jeffrey O Zhang, Bradley Emi, Amir Zamir, Silvio Savarese, Leonidas Guibas, and Jitendra Malik. Learning to navigate using mid-level visual priors. In *CoRL*, pages 791–812, 2020. 4

[41] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *CVPR*, 2020. 3

[42] Chan Hee Song, Valts Blukis, Jonathan Tremblay, Stephen Tyree, Yu Su, and Stan Birchfield. Robospatial: Teaching spatial understanding to 2d and 3d vision-language models for robotics. In *CVPR*, pages 15768–15780, 2025. 2

[43] Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. In *NeurIPS*, pages 251–266, 2021. 4

[44] Peter Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Adithya Jairam Vedagiri IYER, Sai Charitha Akula,

Shusheng Yang, Jihan Yang, Manoj Middepogu, Ziteng Wang, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. In *NeurIPS*, pages 87310–87356, 2024. 2

[45] Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation. In *ICRA*, 2024. 2

[46] Qiucheng Wu, Handong Zhao, Michael Saxon, Trung Bui, William Yang Wang, Yang Zhang, and Shiyu Chang. Vsp: Diagnosing the dual challenges of perception and reasoning in spatial planning tasks for mllms. In *ICCV*, pages 2270–2280, 2025. 2

[47] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. The rise and potential of large language model based agents: A survey. *SCIS*, 68(2):121101, 2025. 1

[48] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *CVPR*, pages 9068–9079, 2018. 4

[49] Peijin Xie, Lin Sun, Bingquan Liu, Dexin Wang, Xiangzheng Zhang, Chengjie Sun, and Jiajia Zhang. Expand vsr benchmark for vllm to expertize in spatial rules. In *AAAI*, pages 8745–8752, 2025. 2

[50] Yifan Xu, Ziming Luo, Qianwei Wang, Vineet Kamat, and Carol Menassa. Point2graph: An end-to-end point cloud-based 3d open-vocabulary scene graph for robot navigation. In *ICRA*, pages 2853–2860. IEEE, 2025. 2

[51] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 7

[52] Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *CVPR*, pages 10632–10643, 2025. 1, 2

[53] Zeyuan Yang, Jiageng Liu, Peihao Chen, Anoop Cherian, Tim K Marks, Jonathan Le Roux, and Chuang Gan. Rila: Reflective and imaginative language agent for zero-shot semantic audio-visual navigation. In *CVPR*, pages 16251–16261, 2024. 1

[54] Hang Yin, Xiuwei Xu, Zhenyu Wu, Jie Zhou, and Jiwen Lu. Sg-nav: Online 3d scene graph prompting for llm-based zero-shot object navigation. In *NeurIPS*, pages 5285–5307, 2024. 1, 2, 3

[55] Bangguo Yu, Hamidreza Kasaei, and Ming Cao. L3mvn: Leveraging large language models for visual target navigation. In *IROS*, pages 3554–3560. IEEE, 2023. 3

[56] Xin Zeng, Haonan Luo, Zihang Wang, Sijia Li, Zhixuan Shen, and Tianrui Li. A continual learning approach for embodied question answering with generative adversarial imitation learning. In *ICASSP*, pages 1–5. IEEE, 2025. 3

[57] Weichen Zhang, Zile Zhou, Xin Zeng, Liu Xuchen, Jianjie Fang, Chen Gao, Jinqiang Cui, Yong Li, Xinlei Chen, and Xiao-Ping Zhang. Open3d-vqa: A benchmark for embodied spatial concept reasoning with multimodal large language model in open space. In *ACMMM*, pages 12784–12791, 2025. 2, 3

[58] Ziang Zhang, Zehan Wang, Guanghao Zhang, Weilong Dai, Yan Xia, Ziang Yan, Minjie Hong, and Zhou Zhao. Dsi-bench: A benchmark for dynamic spatial intelligence. *arXiv preprint arXiv:2510.18873*, 2025. 2

[59] Yong Zhao, Kai Xu, Zhengqiu Zhu, Yue Hu, Zhiheng Zheng, Yingfeng Chen, Yatai Ji, Chen Gao, Yong Li, and Jincai Huang. Cityeqa: A hierarchical llm agent on embodied question answering benchmark in city space. *arXiv preprint arXiv:2502.12532*, 2025. 3

[60] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: a 3d vision-language-action generative world model. In *ICML*, pages 61229–61245, 2024. 2

[61] Duo Zheng, Shijia Huang, Lin Zhao, Yiwu Zhong, and Liwei Wang. Towards learning a generalist model for embodied navigation. In *CVPR*, pages 13624–13634, 2024. 3

[62] Filippo Ziliotto, Tommaso Campari, Luciano Serafini, and Lamberto Ballan. Tango: training-free embodied ai agents for open-world tasks. In *CVPR*, pages 24603–24613, 2025. 1

IndustryNAV: Exploring Spatial Reasoning of Embodied Agents in Dynamic Industrial Navigation

Supplementary Material

A. More Details about IndustryNav

A.1. Scene Layouts

We provide the top-down view layouts in Fig. 8. These scenarios are manually created by five experts using the Unity simulator. To increase diversity, we modify the overall layouts by changing shelf locations, adding safety belts and additional assets, introducing more moving objects and workers, and adjusting the trajectories of dynamic objects and workers. These diverse layouts allow embodied agents to have a wider range of environments and enable a more comprehensive evaluation of their spatial reasoning abilities.

A.2. More Warning Detection Examples

The warning detection examples provided in Fig. 4 offer visual evidence of the our warning detection method's effectiveness, demonstrating that the depth map is a critical component for reliably evaluating whether a safety warning condition exists.

A.3. Diverse Assets

We present the examples of different assets in Fig. 10. Fig. 10 (a) includes a variety of industrial vehicles essential for warehouse operations. The collection features multiple types of forklifts (including reach trucks and counterbalance forklifts) in various color schemes, manual and electric pallet jacks, and forklift attachments such as safety cages and fork extensions. These assets serve as both static obstacles and drivable vehicles within the simulation. Fig. 10 (b) includes dynamic objects and workers. This subset features animated human workers equipped with standard safety gear (hard hats and vests) and autonomous mobile robots depicted transporting inventory. These agents are used to introduce dynamic obstacles into the navigation tasks, which can make the navigation more challenging and realistic. Fig. 10 (c) comprises general warehouse storage items to populate shelves and floor areas. This extensive collection includes large intermodal shipping containers, wooden crates of varying sizes, cardboard boxes, industrial drums (barrels), plastic and wooden pallets, storage bins, and manual transport tools like hand trucks and rolling carts. These objects are arranged to create complex obstacle configurations and realistic visual clutter.

A.4. Embodied Agent Localization

To simplify state representation, we utilize a minimap-based pixel coordinate system for both the embodied agent and the target. The target's location is passed directly using its pixel coordinates. For the embodied agent, its current location is determined by detecting the area of the red cone in the minimap and using the center of this detected area to pinpoint the position.

A.5. Prompts

We provide our navigation prompts for embodied agents in Fig. 11 and Fig. 12.

B. Experiments

B.1. More Failed Action-Reasoning Cases

We provide additional examples of failed action behaviors for the seven evaluated embodied agents in Fig. 13. Analysis of the underlying reasoning process reveals significant deficiencies in the agents' spatial reasoning and decision-making, notably a lack of proactive exploration strategies and insufficient safety awareness. These shortcomings are consistently demonstrated by failures in collision avoidance and unreliable metric distance estimation, indicating that the agents operate purely reactively rather than intelligently anticipating spatial risks.

B.2. More Ablations

B.2.1. Dynamic v.s. Static

To evaluate the influence of dynamic objects or humans in the scenario, we provide the comparison in Fig. 16. From the results, we can see that the Success Ratio and Distance Ratio for three agents is lower in dynamic environment, which demonstrates the presence of moving obstacles (like moving forklifts or walking workers in Fig. 10 (b)) introduces complexity and uncertainty the embodied agents struggle to handle, leading to higher failure rates. Moreover, it can also be observed that the Collision Ratio and the Warning Ratio is higher in the dynamic environment for all agents, especially for GPT-3.5-mini. One possible reason is the agent fail to anticipate or react quickly to the changing positions of dynamic agents. Furthermore, the results also suggest that the agent tends to find better path in static environment with much lower Average Steps, since agents are dealing with dynamic obstacles which requires more micro-

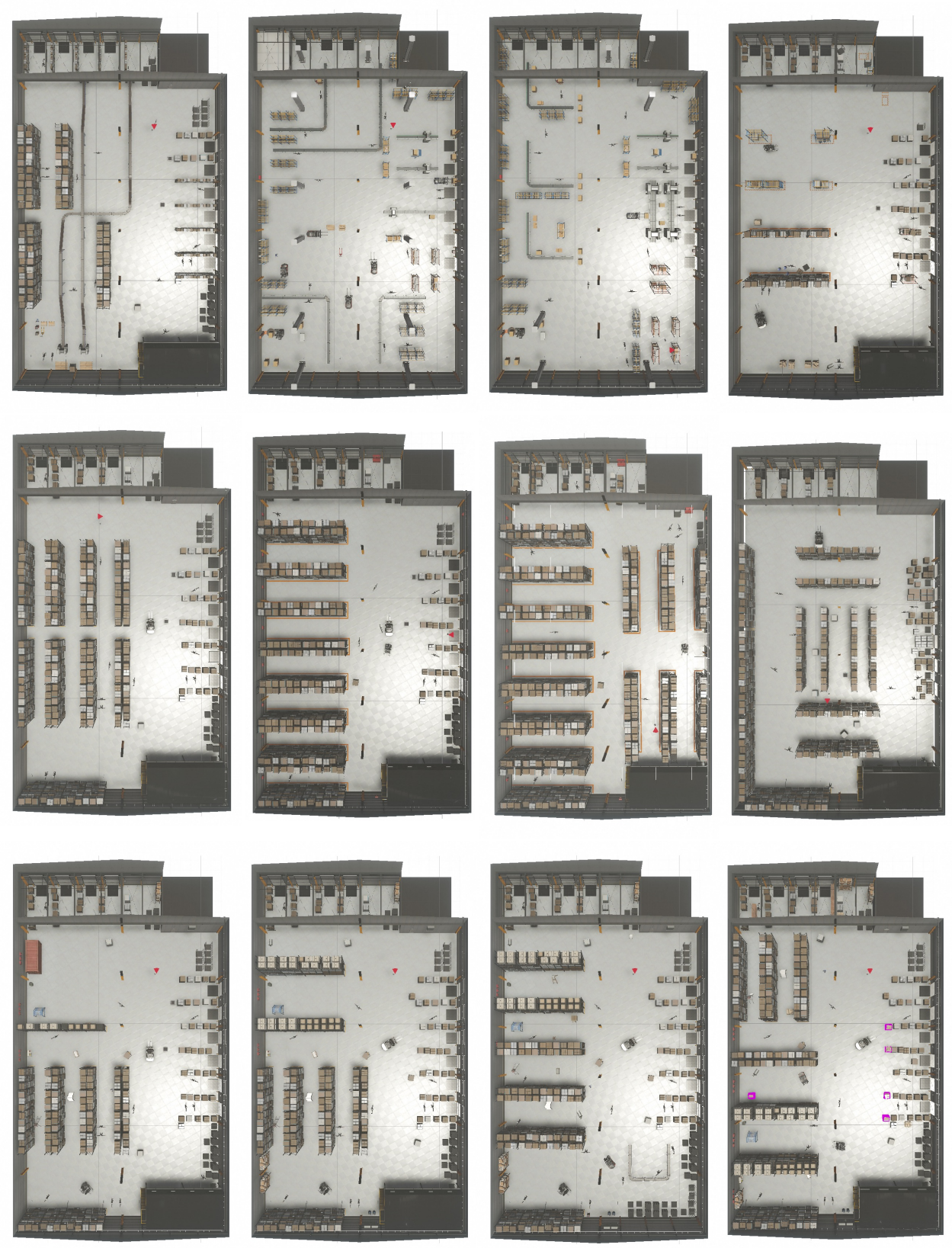


Figure 8. Top-down illustrations of 12 manually created warehouses. These layouts are designed in Unity by five experts using diverse assets and structural modifications.

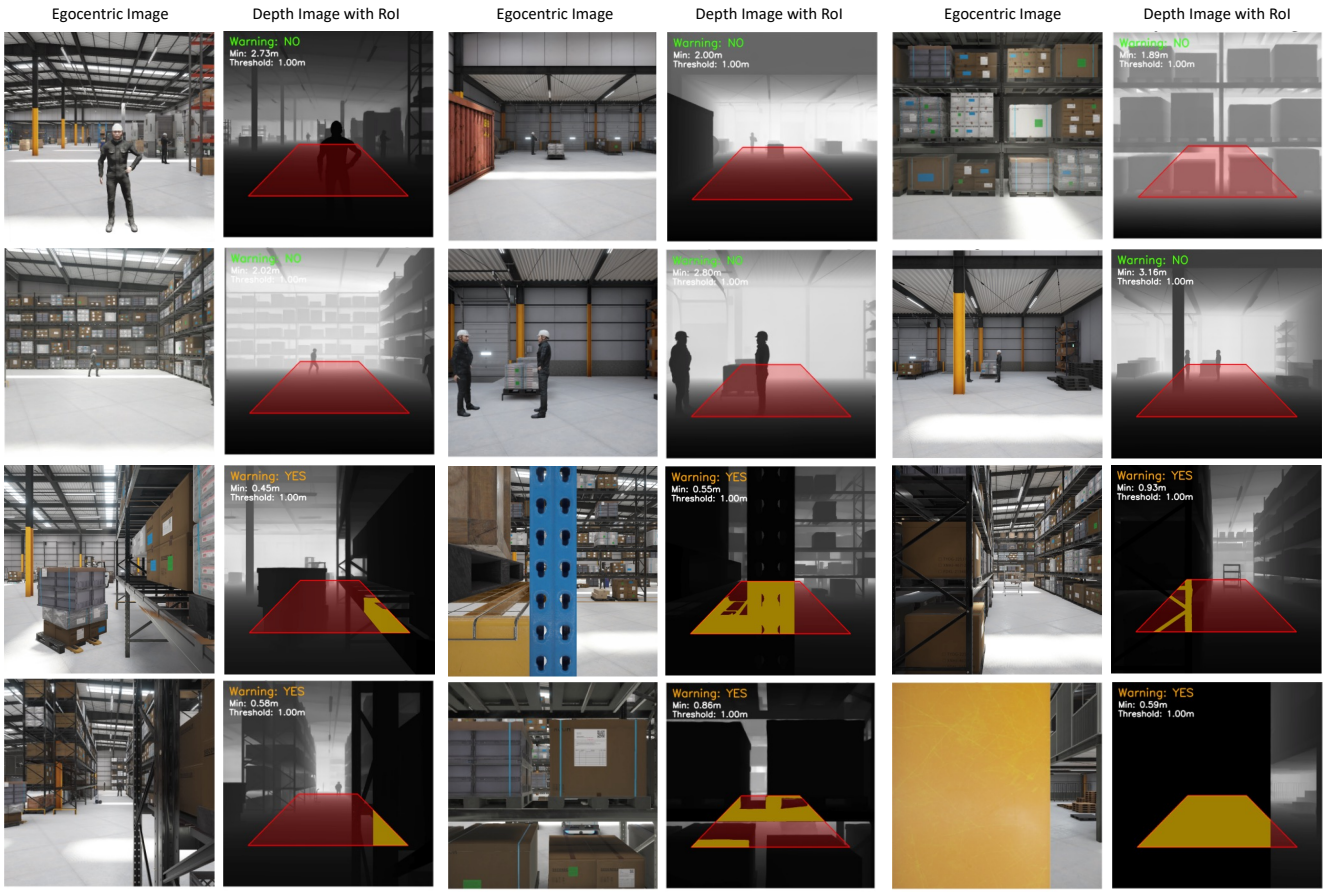


Figure 9. Warning detection illustration.

corrections, and detours, leading to longer completion times compared to the straightforward static path.

These results demonstrate that the navigation task is substantially more challenging in the dynamic industrial scenario. The presence of moving obstacles introduces higher uncertainty and variability, which critically strains the agent's robustness and its ability to maintain high success rates while executing efficient, long-horizon plans.

B.2.2. Effectiveness of Action-State Histories

We provide more results to demonstrate the effectiveness of action-state histories in Fig. 14. the history information is beneficial, leading to noticeable gains in the agent's navigation success rate, overall efficiency, and safety performance.

B.2.3. Necessity of Top-Down View Map

We also include more results in Fig. 15 to illustrate the necessity of top-down view map to the navigation. The results indicate that simply adding a top-down map input is useless to the overall success and safety of these embodied agents. While the map may slightly improve the performance of GPT-5-mini, the performance will drop on other two agents, indicating a fundamental challenge in multi-modal input fusion.

C. Limitations

C.1. Frame Rate Challenges

One practical challenge we faced is that our scenes are quite asset dense (often hundreds of different boxes, pallets, a large variety of other supportive object placements) with complex material reflections, lighting/shadow setups and so on. This inherently limited the ceiling of runtime frame rate despite numerous optimization efforts while under humble GPU resource (regardless of rendering by Vulkan, DirectX or OpenGL). This affected our ability to produce effective asynchronous execution below.

C.2. Lack of Ray Tracing

While the current setup already provides high visual fidelity, the graphics can be significantly more improved if Ray Tracing is enabled. This was omitted due to its drastic increase in computational resource demand from specific GPUs, given the existing frame rate above.

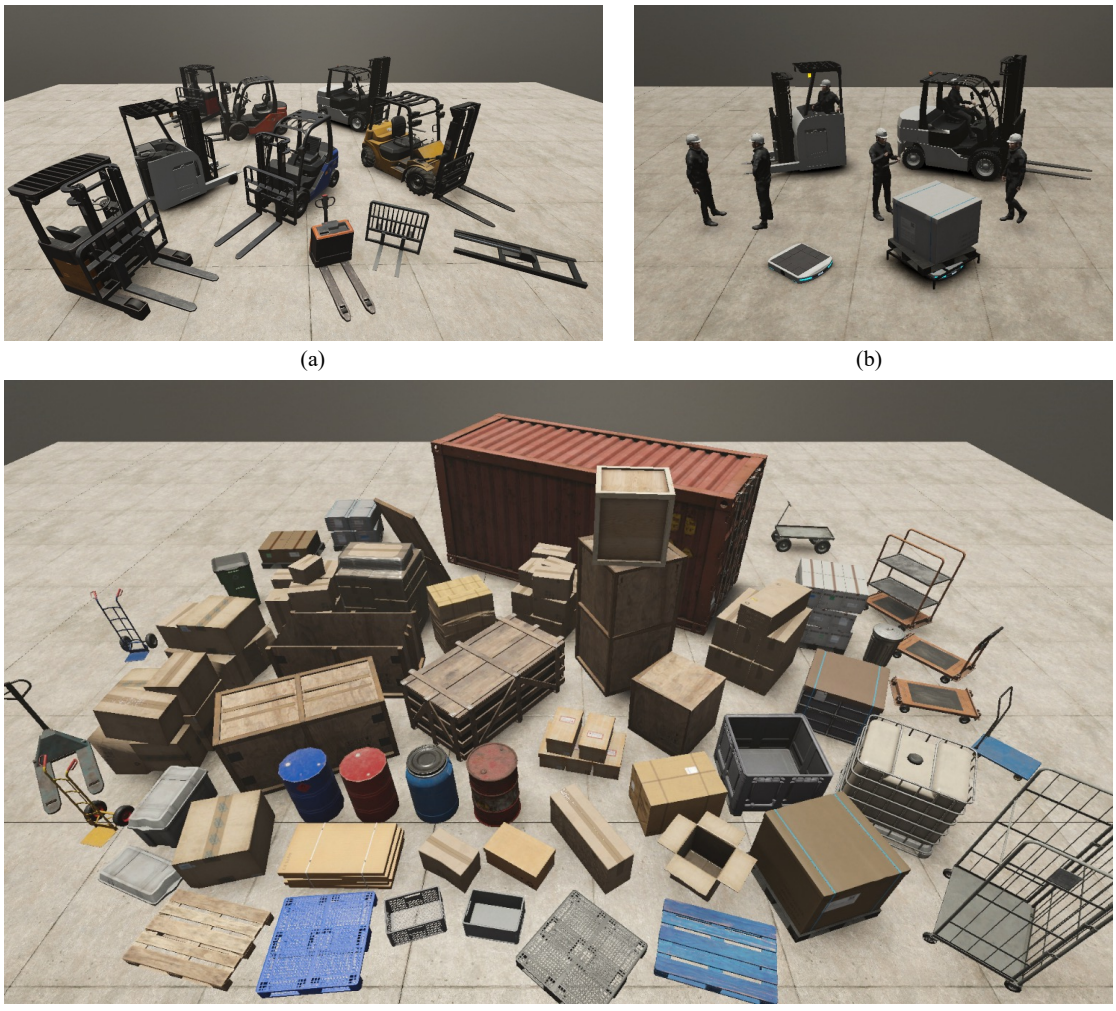


Figure 10. Examples of key dynamic and static components used to build our industrial warehouse environment. (a) Multiple forklift types, (b) movable workers and objects, (c) a wide variety of warehouse assets.

C.3. Lack of Asynchronous Execution

The current MLAgent only supports sequential execution for each action, meaning the environment remains static between steps; this is different from the continuous, dynamic nature of real-world applications.

C.4. Limited Range of Animations

Our collection of animated assets include a variety of motions, such as robots transporting cargo from one place to another, forklifts and reachlifts moving boxes between racks, human worker characters using tablets, walking, chatting, pointing and counting cargo. They currently lack representation of some long tail events, such as injured and/or bleeding workers on the floor, workers climbing onto or coming off of vehicles, worker characters abandoning one hand truck and picking up a different tool, etc. Such asset creation and setup are quite sophisticated to setup at a quality level of sufficient life-like fidelity, hence we omitted them in this work due to resource shortage.

C.5. Lack of Outdoor Scenes

Our collection of scenes do not include outdoor setups, such as clusters of multiple warehouses where robots, workers and vehicles may move from one warehouse to another with or without cargo. Due to a mix of frame rate and complexity reasons, this more expansive setup is currently too challenging given our limited resources, hence we omitted it, in conjunction with activities specifically more associated with those scenarios, such as semi trucks loading and unloading cargo by workers and/or robots, neighborhood delivery trucks for final-mile deliveries, campus security patrols, outside security fences and nearby gardening/vegeta-

Navigation with Egocentric Image and Global Odometry

You are a warehouse navigation agent. At each step, you receive an egocentric camera view and a compact state description. Your task is to reach the target while avoiding all obstacles.

COORDINATE SYSTEM
- Map: +X = East (right), +Y = South (down), -X = West (left), -Y = North (up)
- Headings: $\theta = 0^\circ \rightarrow$ West, $90^\circ \rightarrow$ North, $180^\circ \rightarrow$ East, $270^\circ \rightarrow$ South

CURRENT STATE
- Position: ({`curr_x`}, {`curr_y`})
- Target: ({`target_x`}, {`target_y`})
- Distance to target: {`distance`} px
- Heading: {`theta`}°
- Allowed actions: {`allowed_actions`}

MOVEMENT HISTORY
{`history`}

ACTIONS & DYNAMICS (Step size $\Delta = 34$ px)

Action	Condition / Input	Effect / Result
`turn_right`	-	$\theta \leftarrow (\theta + 90^\circ) \bmod 360^\circ$
`turn_left`	-	$\theta \leftarrow (\theta - 90^\circ) \bmod 360^\circ$
`forward`	Heading 0° (West)	$x \leftarrow x - 34$
	Heading 90° (North)	$y \leftarrow y - 34$
	Heading 180° (East)	$x \leftarrow x + 34$
	Heading 270° (South)	$y \leftarrow y + 34$
`stop`	Dist ≤ 20 px	Terminate episode

DECISION PRIORITY
1. **Check history.** Review recent movements. Avoid repeating failed actions or getting stuck in loops.
2. **Avoid obstacles first.** Use the egocentric view. Never choose an action that collides with walls, shelves, robots, or other objects.
3. **Reduce distance.** Among safe actions, choose the one that moves closer to the target.
4. **Make progress.** If distance hasn't decreased in recent history, consider a different approach.
5. **Stop.** When within 20 px of the target, choose `stop`.

OUTPUT (JSON only)

```
{
  "reasoning":  "Brief logic based on history and obstacles",
  "action":  "<forward|back|strafe right|strafe left|stop>"   (SELECT ONE)
}
```

Figure 11. Navigation prompt for embodied agents with egocentric image, history and global odometry information. Critical simulation parameters, such as the agent's position or the target coordinates, are dynamically inserted into the prompt via placeholder variables enclosed in curly braces, {}.

Navigation with Egocentric Image and Top-Down View Minimap

You are the navigation brain of a warehouse robot. At each step, you receive visual and coordinate information, and your task is to safely reach the target location.

VISUAL INPUTS & MAPPING
- **Egocentric Camera Image:** Used to detect near-field obstacles and immediate collision risks.
- **Top-Down Minimap Image:** Provides the global layout. The robot is a **`RED TRIANGLE`** (tip is facing direction). The target is a **`GREEN DOT`**.

COORDINATE CONTEXT
- **System:** cv2 Pixel Coordinates (resolution: 1024x512).
- **Map Axes:** $+$X $=$ East (right), $+$Y $=$ South (down).

CURRENT STATE OBSERVATIONS
- **Current Position:** ({`curr_x`}, {`curr_y`})
- **Target Position:** ({`target_x`}, {`target_y`})
- **Allowed Actions:** {`allowed_actions`}

GOAL & ACTION SPACE
- **GOAL:** Move safely toward the green dot. Choose `stop` if essentially at the goal (distance $\approx$ few px).
- **ACTION SPACE:** Choose *exactly one* action from the set: `forward`, `back`, `strafe right`, `strafe left`, `stop`.

POLICY (Short and Safe Execution)
1. **Movement:** Use the triangle tip direction on the minimap to decide forward/left/right turns; prefer `forward` if the path is clear.
2. **Alignment:** If the target lies laterally (left/right) of the current tip direction, use `strafe` actions to align the position closer to the target line.
3. **Safety First:** Avoid obstacles visible in the egocentric image. If a collision is imminent, use `strafe` to evade first.
4. **Termination:** If the agent is essentially at the target location, or if no safe progress can be made, choose `stop`.

OUTPUT (STRICT JSON ONLY)

```
{
  "reasoning":  "<1--2 short sentences explaining decision>",
  "action":  "<forward|back|strafe right|strafe left|stop>"  (SELECT ONE)
}
```

Figure 12. Navigation prompt for embodied agents with egocentric image and top-down view minimap information. Critical simulation parameters, such as the agent's position or the target coordinates, are dynamically inserted into the prompt via placeholder variables enclosed in curly braces, {}.

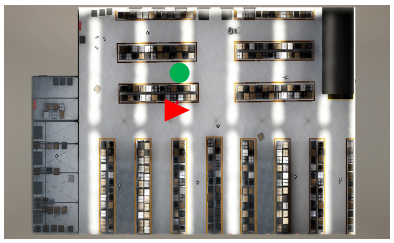
tion, civilian structures and traffic.

D. Future Work

D.1. Frame Rate Optimization

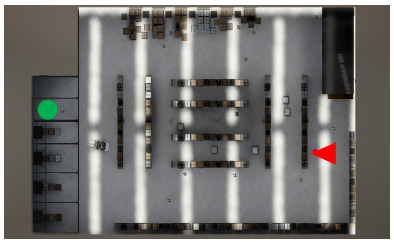
Optimizing runtime frame rate can address many of the issues listed above. Such efforts likely involve preparing lower resolution replacement dummies for an increased percentage of art assets for applying Level-of-Detail (LoD) technology more exhaustively, together with a more comprehensive configuration of conditional real time rendering under camera distances and angles, as well as light and shadow optimizations.

Global Odometry: Position (458,213), θ=180.0°, Target (460, 160), Distance to target: 51.35



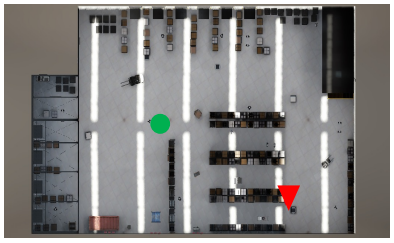

Gemini-2.5-flash **Action**: **turn left** | **Reasoning**: The agent is currently at (458, 213) with a heading of 180 degrees (East). The target is at (460, 160). This means the agent needs to move primarily North (-Y direction). From the current heading, moving forward would take the agent East (+X direction), which is away from the target in the Y coordinate. The history shows a lot of turning right and left, without making significant progress towards the target. In fact, steps 20, 21, 23, 24, 25 seems to be a loop of turning left and right. There are no immediate obstacles in front based on the image. To reach the target, the agent needs to turn towards North (90 degrees). Currently, current heading is 180 degrees(East) i.e. facing to the right. Turning 'left' would change the heading to 90 degrees (North).

Global Odometry: Position (875,318), θ=360.0°, Target (100, 230), Distance to target: 729.33



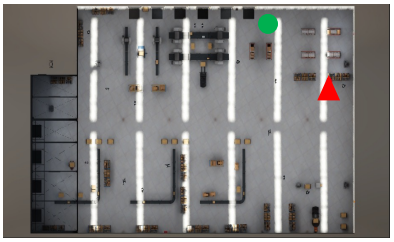

Gemini-2.5-flash **Action**: **forward** | **Reasoning**: The current position is (857, 318) and the target is (100, 230). The current heading is 360.0° (West). We need to move left and forward to reach the target. The current view shows a clear path forward. History shows success with forward/turn left actions. Moving forward will decrease the X coordinate, which is necessary to reach the target's X coordinate of 100.

Global Odometry: Position (744,428), θ=0.0°, Target (400, 256), Distance to target: 390.43



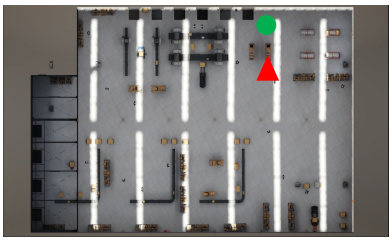

Gemini-2.5-flash **Action**: **turn left** | **Reasoning**: The agent's current heading is West (0 degrees). The target is at (400, 256) and the current position is (744, 428). This means the agent needs to move primarily West (-X) and North (-Y). Currently, the agent is moving forward (West) but the distance hasn't decreased in the last step. The egocentric view shows a large obstacle directly in front. To avoid the obstacle and continue towards the target, a turn is needed. Since the target also has a North component, turning left (North) would be the next logical step after hitting an obstacle while going West. Given the last two movements were 'forward' and the distance didn't change, we need to try a different action to make progress.

Global Odometry: Position (867,166), θ=90.0°, Target (700, 50), Distance to target: 201.70



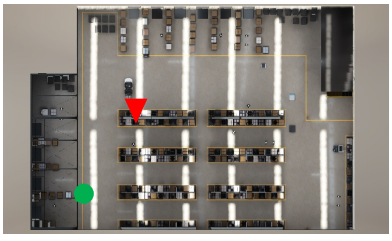
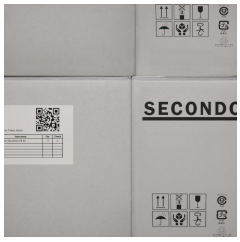

Claude-Sonnet-4.5 **Action**: **forward**| **Reasoning**: History shows consistent progress moving north (θ=90°) from 435px to 203px. Current position (867,166), target (700,50). Need to move west (-167 in X) and north (-116 in Y). Currently heading north (θ=90°), which is correct for reducing Y. Path ahead appears clear with warehouse shelving on sides. Continue forward to keep reducing distance toward target.

Global Odometry: Position (705,150), θ=90.0°, Target (700, 50), Distance to target: 70.18




Claude-Sonnet-4.5 **Action**: **forward**| **Reasoning**: History shows a repeating loop: forward at (705,120), turn left to (701,125), turn right back to (705,120). The agent is stuck. Current position (705,120), target (700,50) - need to go 5px West and 70px North. Heading is 90° (North). Forward would move to (705,86), reducing distance significantly. The egocentric view shows stacked materials/packages ahead, but there appears to be navigable space. Breaking the loop by going forward is essential to make progress toward the target.

Global Odometry: Position (347,230), θ=270.0°, Target (200, 430), Distance to target: 248.21

Claude-Haiku-4.5 **Action**: **forward**| **Reasoning**: History shows agent is stuck in a loop (steps 36-41) oscillating between θ=180° and θ=90°, with position barely changing and distance not decreasing. Current position (347,230), target (200,430). Need to move: -147 X (West) and +200 Y (South). Current heading θ=270° (South) is correct for moving toward target (+Y direction). The camera view shows a mostly clear path ahead with boxes/packages visible on the left side but no obstruction directly in front. Should move forward to make progress toward the target instead of continuing the turning loop.

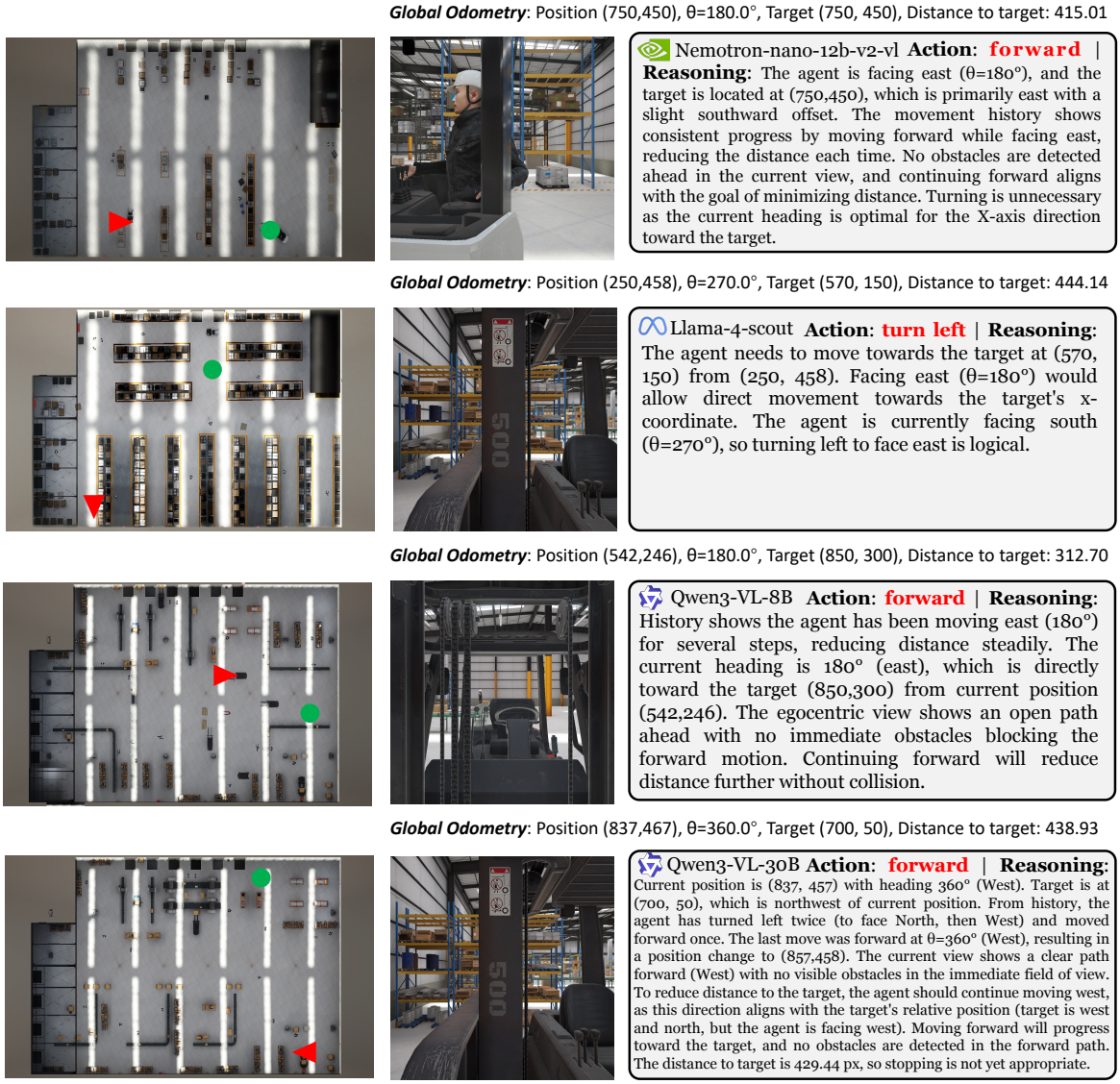


Figure 13. Illustration of incorrect action behaviors of seven embodied agents (from top to bottom): Gemini-2.5-flash, Claude-Sonnet-4.5, Claude-Haiku-4.5, Nemotron-nano, LLaMA-4-Scout, Qwen3-VL-8b, Qwen3-VL-30B.

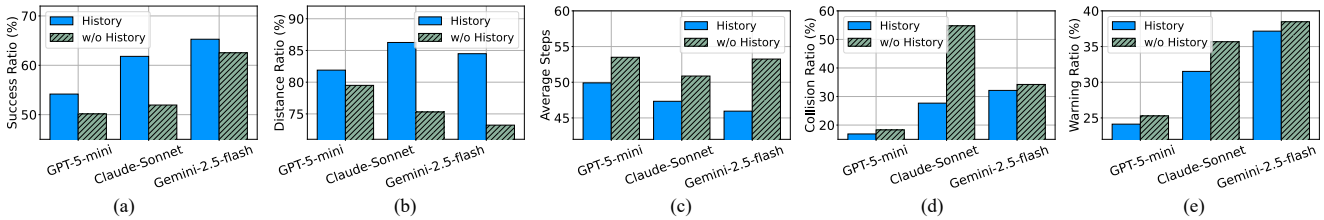


Figure 14. Ablation study of the action-state history. The blue bar represents our full (default) navigation pipeline, while the green bar with texture shows the performance when the history information is removed.

D.2. Proactive Embodied Agent

Our experimental results reveal that current embodied agents lack active exploration and self-correction abilities during the spatial reasoning process, often executing myopic actions. This inherent reactivity limits performance in dynamic and complex environments. To build a proactive

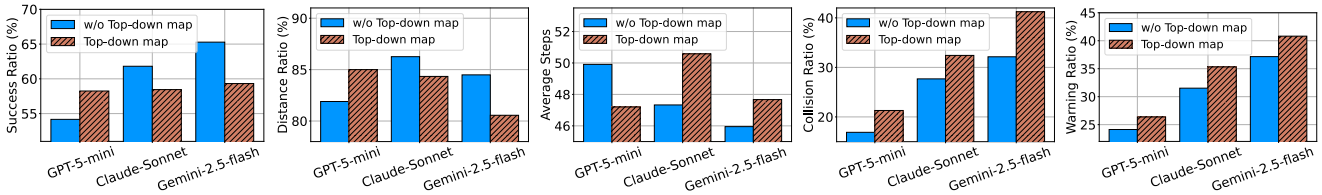


Figure 15. Ablation study of the top-down map. The blue bar represents our default navigation pipeline, while the textured orange bar shows the performance when an extra top-down view map is included.

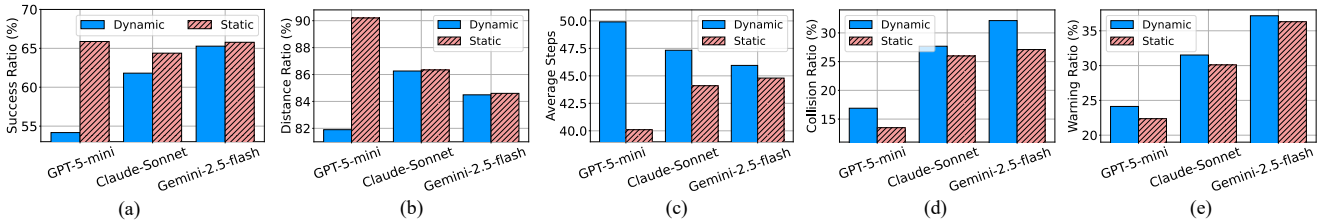


Figure 16. Ablation of dynamic vs. static scenarios. The blue bar represents our default navigation pipeline operating in the dynamic environment (with animation), while the textured red bar shows performance when the dynamic animation is disabled (static scenario).

embodied agent, we plan to employ Reinforcement Learning (RL) to train a policy capable of long-horizon planning. This RL approach will enable the agent to anticipate future states, utilize Monte Carlo methods or similar search techniques to evaluate exploratory actions, and proactively adjust its trajectory to maximize successful task completion and minimize costly re-planning cycles.

D.3. Safety-Aware Embodied Agent

From our experimental results, we find that all the embodied agents struggle with safety issues, frequently resulting in collisions with surrounding obstacles and a failure to make precise distance estimations. This demonstrates a critical lack of robust environmental awareness and metric planning capabilities. To solve this problem, we plan to introduce a novel, safety-centric learning framework that explicitly models both local obstacle avoidance and global metric precision, thereby enabling the agent to learn reliable and inherently safe navigation policies. For even more precise metric perception, additional modalities like depth maps or 3D point clouds can be integrated. This is a vital next step, as these data sources provide the necessary fine-grained spatial measurements required for calculating precise clearances and enabling complex movements.

D.4. Efficient Embodied Agent

All of the evaluated embodied agents possess a large parameter size, making them unsuitable for direct deployment on on-device robots due to power and efficiency constraints. To achieve the requisite efficiency for real-world operation, simply compressing existing large models may not suffice. To build an efficient embodied agent, we plan to design a lightweight and efficient policy architecture, specifically favoring mobile-optimized backbone networks over traditional, heavy structures. This emphasis on architectural efficiency will ensure the agent maintains high throughput with minimal latency, even on embedded systems.