# NavBench: Probing Multimodal Large Language Models for Embodied Navigation

**Yanyuan Qiao**[1] **Haodong Hong**[2 3] **Wenqi Lyu**[4] **Dong An**[5] **Siqi Zhang**[6]
**Yutong Xie**[5] **Xinyu Wang**[4] **Qi Wu**[4*]
[1]Swiss Federal Institute of Technology Lausanne (EPFL)
[2]The University of Queensland [3]CSIRO Data61 [4]The University of Adelaide
[5]Mohamed bin Zayed University of Artificial Intelligence [6]Tongji University

Project Website

## Abstract

Multimodal Large Language Models (MLLMs) have demonstrated strong generalization in vision-language tasks, yet their ability to understand and act within embodied environments remains underexplored. We present NavBench, a benchmark to evaluate the embodied navigation capabilities of MLLMs under zero-shot settings. NavBench consists of two components: (1) navigation comprehension, assessed through three cognitively grounded tasks including global instruction alignment, temporal progress estimation, and local observation-action reasoning, covering 3,200 question-answer pairs; and (2) step-by-step execution in 432 episodes across 72 indoor scenes, stratified by spatial, cognitive, and execution complexity. To support real-world deployment, we introduce a pipeline that converts MLLMs' outputs into robotic actions. We evaluate both proprietary and open-source models, finding that GPT-4o performs well across tasks, while lighter open-source models succeed in simpler cases. Results also show that models with higher comprehension scores tend to achieve better execution performance. Providing map-based context improves decision accuracy, especially in medium-difficulty scenarios. However, most models struggle with temporal understanding, particularly in estimating progress during navigation, which may pose a key challenge.

## 1 Introduction

Multimodal Large Language Models (MLLMs) [1, 2, 3] have achieved impressive performance across a wide range of vision-language tasks, demonstrating strong cross-modal reasoning and zero-shot generalization. These models excel at answering visual questions [4], interpreting videos [5], and performing complex multimodal reasoning [6]. As their capabilities expand, a central question emerges: do these models truly understand how to act in the physical world, or are they simply adept at processing static inputs?

Recent work has begun to explore MLLMs' potential in embodied tasks by evaluating their spatial reasoning in 3D environments [7, 8]. However, these tasks primarily focus on perception and passive scene understanding, without assessing the model's ability to make decisions or take actions. In comparison, navigation is a core embodied task that involves interpreting natural language instructions, analyzing visual observations, and making a sequence of decisions to reach a goal. Although navigation plays a crucial role in real-world applications, it remains relatively underexplored in the context of MLLMs. Traditional embodied navigation benchmarks, such as Room-to-Room (R2R) [9]

*Corresponding author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

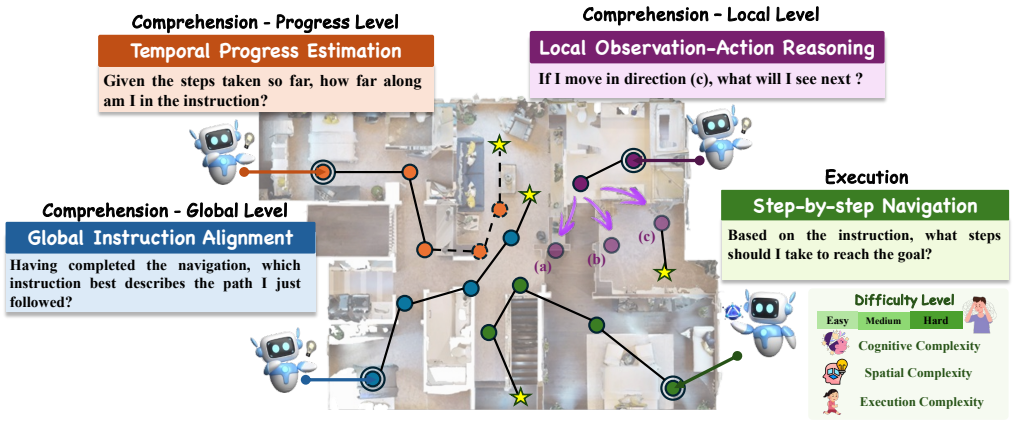


Figure 1: NavBench evaluates MLLMs across three comprehension tasks and a step-by-step execution task, assessing their ability to understand navigation behavior, track progress, reason about observation and action, and act accordingly. The step-by-step navigation is assessed from different difficulty levels, which is defined by cognitive, spatial, and execution complexity.

and ObjectNav [10], were developed prior to the emergence of foundation models. These benchmarks rely on task-specific supervision and often reduce evaluation to final success rates, providing limited insight into whether a model genuinely understands the navigation behavior. In many cases, an agent may reach the goal by exploiting dataset biases or learning shortcuts, without correctly grounding the instruction or following the intended path.

Similar to how humans acquire embodied skills by first understanding a task and then learning to execute it, evaluating the embodied capabilities of generalist MLLMs also requires examining two fundamental aspects. First, can the model comprehend what a navigation behavior represents, such as identifying the intent behind a completed trajectory? Second, can it act autonomously to complete a navigation task, making step-by-step decisions in unfamiliar environments? Furthermore, navigation tasks in real-world environments can vary significantly in difficulty due to differences in spatial layout, instruction complexity, and required decision-making steps. For example, navigating across multiple rooms with ambiguous instructions poses greater challenges than following simple step-by-step commands in a single hallway. However, most existing benchmarks treat all navigation episodes equally difficult, failing to capture this essential variation.

To fill these gaps, we introduce **NavBench**, a benchmark designed to systematically evaluate MLLMs in embodied navigation under zero-shot settings. NavBench decomposes the evaluation into two complementary components: *Navigation Comprehension*, which assesses whether a model understands and aligns with intended navigation behavior, and *Navigation Execution*, which evaluates the model's ability to make accurate step-by-step decisions. To reflect real-world variability, NavBench incorporates a fine-grained difficulty classification based on spatial, cognitive, and execution complexity. In addition, it provides a deployable real-world navigation pipeline to bridge the gap between simulation and practical embodiment.

**First**, for navigation behavior comprehension, inspired by cognitive studies of human spatial reasoning [11], NavBench introduces three fine-grained evaluation tasks designed to assess distinct reasoning capabilities at three levels: global, progress, and local. It includes 3,200 question-answer pairs. Specifically, *Global Instruction Alignment* evaluates the model's ability to match a given trajectory with the most appropriate instruction. The candidate instructions are designed with subtle semantic differences, such as variations in directional cues and landmark descriptions, to encourage genuine spatial reasoning. *Temporal Progress Estimation* measures temporal-contextual awareness by requiring the model to infer progress within multi-step instructions based on a partial trajectory. *Local Observation-Action Inference* evaluates the model's ability to reason about the spatial consequences of individual actions by either predicting the future observation given an action or identifying the action that caused a visual transition. Together, these tasks provide a comprehensive framework for assessing global semantic reasoning, temporal understanding, and local spatial inference in navigation.

**Second**, NavBench introduces a fine-grained difficulty classification with three levels: easy, medium, and hard, based on cognitive, spatial, and execution complexity. This allows detailed analysis of

models' generalization and decision-making performance across varying levels of difficulty. The benchmark includes 432 navigation cases across 72 scenes.

**Finally**, to bridge the gap between simulator-based evaluation and real-world deployment, we design a practical navigation pipeline that connects MLLM outputs to executable actions on real robots. This pipeline includes a waypoint selection module, an MLLM-based navigator, and a low-level controller, demonstrating the deployability of our framework in physical environments.

We evaluate both closed-source and open-source MLLMs on NavBench. While GPT-4o currently achieves the best overall performance, we observe that lightweight models such as Qwen2.5-VL-7B are capable of reliably completing easy navigation tasks. Notably, this trend is also reflected in our real-world deployment experiments, suggesting that NavBench may serve as a practical tool for analyzing the embodied capabilities of both general and resource-efficient MLLMs. Furthermore, our results suggest several notable trends: (1) comprehension and execution abilities appear to be closely related, (2) temporal reasoning may pose a persistent challenge for current models, and (3) compact open-source models can, under certain conditions, approach the performance of proprietary ones, indicating their potential utility in practical settings.

In summary, our main contributions are as follows: (1) We introduce *NavBench*, a benchmark for evaluating MLLMs in embodied navigation under zero-shot settings. (2) We decompose the evaluation into two components: *Navigation Comprehension*, with tasks targeting spatial, temporal, and local reasoning, and *Navigation Execution*, which assesses decision-making across difficulty levels. (3) We develop a deployment pipeline that maps MLLM outputs to real-world robot actions. (4) We perform a detailed evaluation and analysis of both closed-source and open-source MLLMs, uncovering trends in their reasoning and execution performance across embodied tasks.

## 2 Related Work

**Benchmarks for MLLMs** Recent progress in Multimodal Large Language Models (MLLMs)[1, 12, 13, 14, 15] has driven the development of benchmarks assessing visual understanding and cross-modal reasoning. Early efforts such as VQA[4], GQA [16], OK-VQA [17], and TextVQA [18] focus on specific tasks like factual or commonsense question answering. More recent benchmarks including MME [19], MMBench [20], MM-Vet [21], and MathVista [6] aim for broader coverage, evaluating perception and reasoning across diverse domains. However, these mainly target static tasks and do not reflect MLLMs' ability to act in dynamic environments. To bridge this gap, some recent work has begun evaluating spatial reasoning in embodied settings. SpatialBench [7], ScanReason [22], and VSI-Bench [8] assess 3D spatial understanding using panoramas, semantic layouts, or textual scene descriptions. While insightful for embodied perception, they remain limited to passive tasks and do not assess decision-making or sequential interaction. In parallel, traditional embodied navigation benchmarks such as R2R[9], REVERIE[23], and ObjectNav [10] have long been used to test instruction-following agents. However, they were designed for fully supervised settings and mainly evaluate success rates without probing intermediate reasoning. Although REVERIE increases instruction abstraction, it retains similar path lengths and decision complexity, limiting its capacity to reveal behavioral differences. More recently, Wang *et al.* [24] proposed a fine-grained evaluation framework for instruction understanding in VLN via multiple-choice questions, offering interpretability beyond end-to-end metrics. Still, their setup is restricted to small supervised models and lacks real-world deployment and zero-shot inference. To the best of our knowledge, no existing benchmark offers a comprehensive evaluation of MLLMs in embodied navigation that jointly considers instruction understanding, sequential decision-making, difficulty stratification, and real-world transferability.

**Embodied Navigation** Embodied navigation tasks require an agent to reach a goal location within an environment, guided by a description such as an image [25, 26], object [10, 27], or natural language instruction [9, 28, 29]. Among these, language-guided navigation has attracted significant attention for its potential to facilitate intuitive human-robot interaction. Researchers have explored diverse instruction formats, including step-by-step [9, 30], dialog-based [31], and goal- or intention-oriented instructions [23, 32]. Traditional approaches train navigation policies using annotated datasets, incorporating modules to improve object relation understanding [33, 34, 35], vision-language alignment [36, 37, 38], memory [39, 40, 41], and spatial reasoning [42, 43, 44]. While effective on benchmarks, these methods often suffer from limited generalization due to dataset biases [45, 46, 47]. To mitigate this, recent work turns to MLLMs for zero-shot embodied navigation, leveraging their generalization abilities. Some use MLLMs to localize goal-relevant regions [48, 49, 50], while others

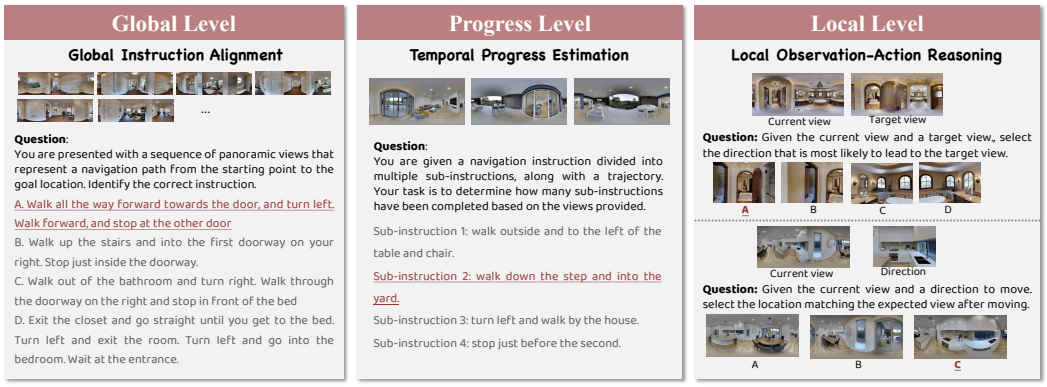


Figure 2: Illustration of the Navigation Comprehension task.

employ prompt-based guidance for instruction following [51, 52, 53, 54]. These approaches reduce reliance on task-specific training but still lack fine-grained evaluation: most benchmarks focus solely on final success rates, offering limited insight into the model's reasoning process. To address this, we introduce NavBench, a benchmark that systematically evaluates both the reasoning and execution capabilities of MLLMs in embodied navigation.

# 3 Benchmark Design

## 3.1 Task Formulation

We evaluate the navigation capabilities of MLLMs by decomposing the task into two core components: *Navigation Comprehension*, which assesses the understanding of navigation behavior, and *Navigation Execution*, which focuses on step-by-step decision making.

**Navigation Comprehension** It investigates whether the model can understand and reason about implicit navigation behaviors, including aligning instructions with trajectories, estimating progress along a plan, and predicting the spatial consequences of actions. These tasks span different reasoning levels (*global*, *progress*, and *local*) and serve as diagnostic probes for navigation understanding. Illustrations of the three comprehension tasks are shown in Figure 2[2].

- Global Level – Global Instruction Alignment: Given a navigation trajectory and several candidate instructions, the model is required to determine which instruction aligns with the executed path. This task tests the model's understanding of the overall intent and structural coherence of the navigation behavior.
- Progress Level – Temporal Progress Estimation: Provided with a partial trajectory and a list of segmented sub-instructions, the model must identify the sub-instruction that was most recently completed. This evaluates the model's capacity to monitor task progress and comprehend the temporal structure of instructions.
- Local Level – Local Observation-Action Reasoning: To evaluate the model's ability to reason about the spatial consequences of individual actions. We design two variants: (1) Future-Observation Prediction – the model observes the current view and an action, and selects the correct resulting view. (2) Future-Action Prediction – the model observes two consecutive views and must identify the action that caused the transition.

**Navigation Execution** It examines whether an MLLM can make accurate, step-by-step movement decisions in an embodied environment based on the current observation and instruction. We conduct this evaluation in a zero-shot setting [54] within the Matterport3D simulator [55], categorizing tasks into three difficulty levels (easy, medium, and hard) to assess performance. To ensure a fair and standardized evaluation protocol, we evaluate MLLMs via viewpoint selection rather than low-level action prediction (*e.g.*, turning or moving forward). This abstraction, consistent with prior embodied navigation benchmarks [9, 23], allows us to focus on high-level semantic reasoning grounded in language and vision, while avoiding the confounding variability introduced by continuous control. It also facilitates zero-shot evaluation and comparability across different models. Notably, while our

[2]The questions in the figure are slightly simplified for clarity and brevity.

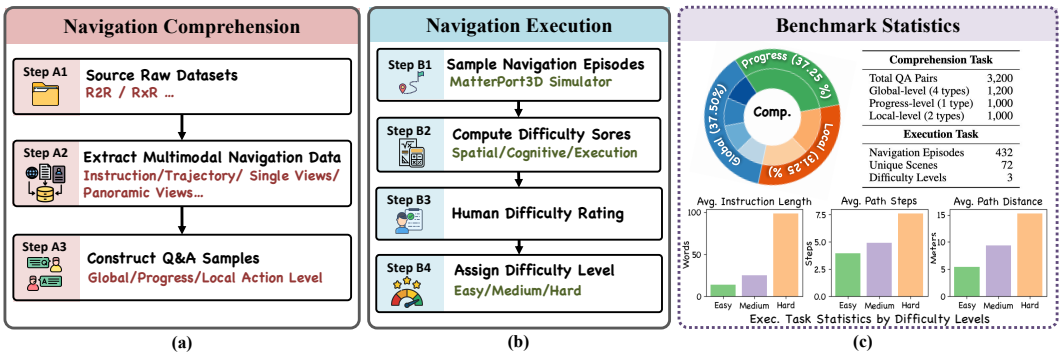


Figure 3: **NavBench construction pipeline and statistics.** (a) QA generation for comprehension tasks at global, progress, and local levels. (b) Execution pipeline combining automatic difficulty scoring and human ratings. (c) Benchmark statistics, including comprehension (comp.) task distribution, QA counts, and execution statistics (*e.g.*, instruction length, steps, distance).

simulator setup centers on abstracted decision-making, Section 4 illustrates how this framework can be extended to real-world navigation by converting viewpoint selection into low-level control.

Specifically, at each step, the model receives the current panoramic observation, the natural language instruction, and a list of candidate navigable viewpoints. The model must select the next location to move to, thereby executing the instruction step-by-step until the goal is reached. Formally, at each step $t$ of a navigation episode, the MLLM receives an instruction $\mathbf{x} = \{w_1, w_2, ..., w_L\}$ of length $L$, a set of candidate navigable observation viewpoints $\mathbf{O}_t = \{o_t^1, o_t^2, ..., o_t^N\}$, and optional context $\mathbf{C}_t$ (such as navigation history or previous actions). The agent must select an action $a_t$ corresponding to one of the navigable directions:

$$\mathbf{a}_t = \text{MLLM}(\mathbf{x}, \mathbf{O}_t, \mathbf{C}_t)\,. \tag{1}$$

This decision process may involve reasoning about the instruction, interpreting the current view, leveraging prior context, and anticipating the result of each candidate action.

### 3.2 Dataset Construction

**Data Sources** NavBench is constructed by reorganizing and enriching fine-grained navigation data with multimodal observations to enable zero-shot evaluation of MLLMs. We start by collecting instruction-trajectory pairs from multiple embodied navigation benchmarks, including R2R [9], RxR [30], GEL-R2R [56], and FGR2R [57]. These datasets serve as annotation sources, but do not include the visual inputs needed for multimodal reasoning. To address this gap, we use the Matterport3D simulator to extract both panoramic and single-viewpoint RGB images aligned with navigation trajectories. The image extraction process involves traversing agent paths, sampling intermediate viewpoints, and rendering corresponding visual observations. All visual and textual data are then organized into a unified structure that supports multiple reasoning tasks and enables consistent QA generation across comprehension and execution settings. Figure 3 shows the overall benchmark construction pipeline.

**Statistics** We report statistics in Figure 3(c), including distribution of comprehension subtasks and coverage of scenes and episodes in execution. These statistics reflect the scale and diversity of the benchmark across reasoning levels and scenes.

#### 3.2.1 Question-and-Answer Pairs Collection

We design three diagnostic tasks targeting global alignment, temporal progress estimation, and local spatial and action reasoning. In total, we collect 3,200 question-and-answer pairs to evaluate comprehension capacity in embodied navigation.

**Global Instruction Alignment** To evaluate MLLMs' ability to align spatial trajectories with semantically consistent instructions, we construct a multiple-choice dataset comprising 1,200 examples. Each example consists of a panoramic trajectory and five candidate instructions, including one ground-truth

and four distractors. The distractors are generated using four perturbation strategies: (1) *Basic*: random instructions sampled from unrelated trajectories, testing global relevance; (2) *Directional replacements*, where spatial terms (*e.g.*, "left", "north") are substituted using POS tagging via NLTK, probing directional grounding; (3) *Object replacements*, where noun phrases are replaced with unrelated landmarks drawn from an external landmark-annotated dataset [56], evaluating object-trajectory grounding; (4) *Shuffled segments*, where human-annotated sub-instructions [57] are permuted to disrupt temporal structure while preserving grammaticality. Each instruction set is randomly ordered and paired with a panoramic trajectory composed of viewpoint sequences and movement annotations. The design promotes multimodal spatial reasoning and reduces reliance on superficial cues.

**Progress Estimation** This task is designed to evaluate a model's ability to perform temporal reasoning and monitor execution progress during navigation. Each full navigation instruction is segmented into a sequence of sub-instructions, and each sub-instruction is aligned with a corresponding portion of the agent's trajectory. We leverage fine-grained annotations [57], which provide this alignment between individual sub-instructions and the associated panoramic viewpoints traversed during execution. To construct evaluation examples, we truncate the trajectory at intermediate points that mark the end of specific sub-instructions. The model is presented with the truncated panoramic trajectory along with the full list of sub-instructions, and is required to predict the index of the last completed one. To ensure data quality and minimize ambiguity, we applied a combination of automatic filtering and manual validation to retain instruction-path pairs with well-defined temporal boundaries (details in Appendix). In total, we collect 1,000 such examples for evaluation.

**Local Observation-Action Reasoning** We design two multiple-choice reasoning tasks to evaluate a model's capacity for local spatial and action reasoning inference. Both tasks present ambiguous scenarios that require fine-grained visual discrimination and understanding of plausible transitions. In Future-Observation Prediction, the model receives a current view and an action, and must choose the correct resulting view from a set of candidates. In Future-Action Prediction, the model observes two consecutive views and selects the action that best explains the transition. For both tasks, distractors are carefully sampled from nearby observations or visually similar actions to ensure ambiguity and challenge. We collect 500 examples for each format, yielding a total of 1,000 samples. All questions are formatted as multiple-choice queries to ensure consistency across evaluation tasks.

### 3.2.2 Navigation Episodes Collection

We sample 432 navigation cases from 72 unique scenes in the Matterport3D simulator [55]. To systematically assess the difficulty of each case, we define a composite complexity score across three orthogonal dimensions: *spatial*, *cognitive*, and *execution* complexity. Each dimension is derived from structural properties of the environment or linguistic cues in the instruction, following the methodology inspired by [58, 59]. In addition, human evaluation is conducted to further support and validate the difficulty classification process.

**Spatial Complexity** It quantifies the geometric and topological challenges of a navigation trajectory. We consider four features: (1) total path length $d$, (2) standard deviation of turn angles $\theta$, (3) vertical range $z$ as a proxy for elevation change, and (4) 2D spatial area $A$ covered by the path. A binary indicator $\mathbb{I}(z > 1.5)$ is included to capture significant elevation changes such as floor transitions. These features are computed from agent poses and scene connectivity data. The spatial complexity score is defined as:

$$\Phi_{\text{spatial}} = \alpha_1 \cdot \log(1 + d) + \alpha_2 \cdot \log(1 + \theta) + \alpha_3 \cdot \mathbb{I}(z > 1.5) + \alpha_4 \cdot \log(1 + A). \tag{2}$$

**Cognitive Complexity** It reflects the linguistic difficulty of navigation instructions. We extract five features using dependency parsing: (1) instruction length $L$, (2) number of verbs $V$, (3) number of spatial terms $S$ (*e.g.*, *left*, *upstairs*), (4) number of landmark mentions $M$ (*e.g.*, *kitchen*), and (5) number of subordinate clauses $C$ (*e.g.*, `relcl`, `advcl`). The cognitive complexity score is defined as:

$$\Phi_{\text{cognitive}} = \beta_1 \cdot \log(1 + L) + \beta_2 \cdot \log(1 + V) + \beta_3 \cdot \log(1 + S) + \beta_4 \cdot \log(1 + M) + \beta_5 \cdot C. \tag{3}$$

**Execution Complexity** It measures the behavioral effort required to complete the navigation. We consider: (1) number of steps $N$, (2) number of turns $T$, (3) floor change indicator $F$, and (4) number of decision points $D$. The score is computed as:

$$\Phi_{\text{execution}} = \gamma_1 \cdot \log(1 + N) + \gamma_2 \cdot \log(1 + T) + \gamma_3 \cdot F + \gamma_4 \cdot D. \tag{4}$$

**Normalization** Each raw complexity score $\Phi$ is normalized to the range $[1, 9]$ using a non-linear mapping:

$$\hat{\Phi} = \text{round}\left(1 + 8 \cdot \frac{\log(1+\Phi) - \log(1+\Phi_{\min})}{\log(1+\Phi_{\max}) - \log(1+\Phi_{\min})}, 2\right). \tag{5}$$

The weights $\alpha$, $\beta$, and $\gamma$ are empirically set to balance the contribution of each factor.

**Human Evaluation** To complement the automatic scoring, we conducted a human evaluation to validate our difficulty annotations. A group of annotators independently rated each case along the three defined dimensions, using a 1–9 scale with detailed guidelines aligned to our scoring criteria. Further details are provided in the Appendix.

**Difficulty Categorization** Based on the final scores, each case is categorized into one of three levels, as illustrated in Figure 4:

- Easy (score 1–3): Short paths with simple instructions, few steps, minimal spatial reasoning, and clear landmarks.
- Medium (score 4–6): Instructions with moderate length, multiple landmarks or spatial phrases, and medium-length paths.
- Hard (score 7–9): Long trajectories guided by complex multi-step instructions, often involving floor transitions and multiple spatial references.

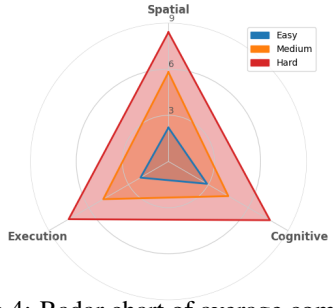


Figure 4: Radar chart of average complexity scores across cognitive, spatial, and execution dimensions for different difficulty levels.

## 4 Real-World Deployment Pipeline

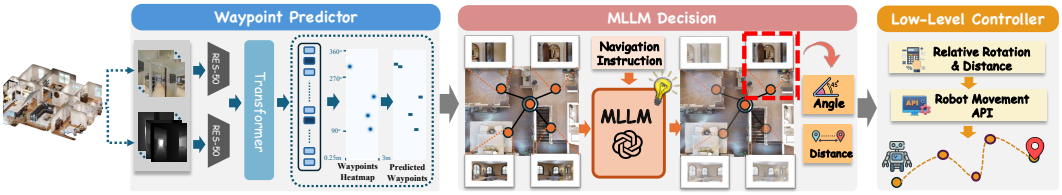


Figure 5: **Overview of the real-world embodied navigation pipeline.**

To demonstrate the real-world feasibility of MLLM-guided embodied navigation, we implement a modular pipeline that complements our benchmark evaluation, as illustrated in Figure 5. It consists of three modules: (1) a *Waypoint Predictor* that extracts RGB and depth inputs to generate candidate waypoints, (2) an *MLLM Decision Module* that selects the most goal-aligned waypoint, and (3) a *Low-Level Controller* that translates the selected waypoint into motion commands for execution on a physical robot. The system is deployed on a dual-arm mobile robot equipped with an RGB-D camera and evaluated in real indoor environments. More details are provided in the Appendix.

## 5 Evaluation on NavBench

### 5.1 Settings

**Models** We evaluate both proprietary and open-source MLLMs widely adopted in recent research. Proprietary models include GPT-4o, GPT-4o-mini, Gemini-2.0-flash. Open-source models include InternVL2.5-2B/8B [60], Qwen2.5-VL-3B/7B [61], LLaVA-OneVision-7B [62], LLaVA-Next-7B, and Llama3.2-Vision-11B [63].

**Implementation Details** Proprietary models are accessed via APIs, while open-source models are deployed using `vLLM` [64] and `lmdeploy` [65] on a single NVIDIA A6000 GPU (48GB). Simulator-based evaluations are conducted in the Matterport3D Simulator [55], built on high-resolution RGB-D scans of real indoor environments such as homes and offices. It provides realistic visual inputs and discrete agent movement within a 3D mesh, making it a standard testbed for embodied navigation. For real-world deployment, we integrate our pipeline with a dual-arm composite mobile robot equipped with an Intel RealSense D435 camera and a Water Drop 2 wheeled base. All physical experiments are conducted in a controlled indoor lab to assess robustness and feasibility.

Table 1: Performance comparison on **Navigation Comprehension** and **Execution**.

| Model | Navigation Comprehension | | | | Navigation Execution | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Global | Progress | Local | Comp. Avg | Easy | | Medium | | Hard | | Exec. Avg |
| | Accuracy | | | | SR | SPL | SR | SPL | SR | SPL | |
| Chance Level (Random) | 19.33 | 25.4 | 29.34 | 24.65 | 16.41 | 9.57 | 7.17 | 3.72 | 7.33 | 4.99 | 8.19 |
| ***VLN-Bench (tiny) Performance*** | | | | | | | | | | | |
| †Human Level | 88.33 | 79.00 | 85.00 | 84.11 | 91.67 | 88.68 | 87.50 | 81.53 | 75.00 | 65.17 | 81.59 |
| †GPT-4o | 51.67 | 45.00 | 63.00 | 53.89 | 66.08 | 49.01 | 43.79 | 36.44 | 25.00 | 20.11 | 40.07 |
| †Qwen2.5-VL-7B | 36.67 | 32.00 | 47.00 | 38.56 | 46.25 | 35.59 | 25.27 | 18.93 | 12.50 | 5.93 | 24.41 |
| ***Closed Models*** | | | | | | | | | | | |
| GPT-4o | 51.33 | 42.90 | **65.80** | 53.34 | **67.36** | **54.31** | 41.67 | 35.71 | **27.78** | **21.15** | **41.33** |
| GPT-4o-mini | 50.33 | 29.90 | 59.03 | 46.42 | 46.53 | 40.44 | 28.47 | 24.90 | 15.28 | 12.29 | 27.99 |
| Gemini-2.0-flash | **79.68** | 40.30 | 32.00 | 50.66 | 61.81 | 45.05 | **46.53** | **39.08** | 25.69 | 16.64 | 39.13 |
| o4-mini | 76.67 | **43.60** | 58.70 | **59.66** | 47.92 | 44.77 | 26.39 | 22.70 | 15.97 | 10.13 | 28.98 |
| ***Open-Source Models*** | | | | | | | | | | | |
| InternVL2.5-2B | **67.25** | 23.40 | 11.25 | 33.97 | 25.69 | 25.29 | 6.94 | 6.68 | 7.64 | 5.86 | 13.02 |
| Qwen2.5-VL-3B | 43.83 | 21.30 | **50.63** | 38.59 | 23.61 | 17.52 | 12.50 | 8.88 | 10.26 | 5.24 | 13.00 |
| InternVL2.5-8B | 62.75 | 28.50 | 28.12 | 39.79 | 28.47 | 28.19 | 7.66 | 7.42 | 7.64 | 6.18 | 14.26 |
| Qwen2.5-VL-7B | 57.58 | **31.20** | 47.00 | **45.26** | **41.67** | **32.55** | **22.92** | **17.43** | 10.42 | 5.67 | **21.77** |
| LLaVA-OneVision-7B | 31.17 | 26.60 | 39.00 | 32.26 | 31.25 | 17.64 | 15.58 | 7.80 | **15.02** | **7.84** | 15.86 |
| LLaVA-Next-7B | 38.33 | 27.40 | 28.50 | 31.41 | 27.08 | 25.95 | 11.81 | 7.69 | 7.64 | 6.07 | 14.54 |
| Llama3.2-Vision-11B | 36.00 | 23.40 | 29.10 | 14.75 | 27.08 | 25.90 | 10.42 | 9.19 | 10.02 | 7.60 | 15.04 |

*Note:* **Dark teal** and **light teal** indicate the top-performing closed and open-source models per column.
† indicates results evaluated on the NavBench (tiny) subset.

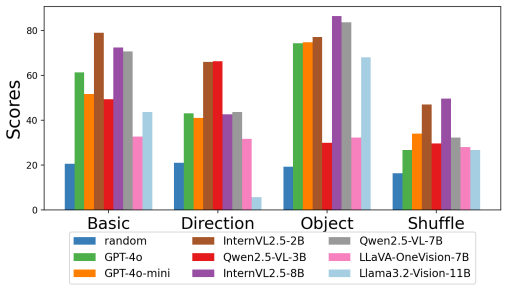


Figure 6: Model Performance under Different Instruction Perturbations.

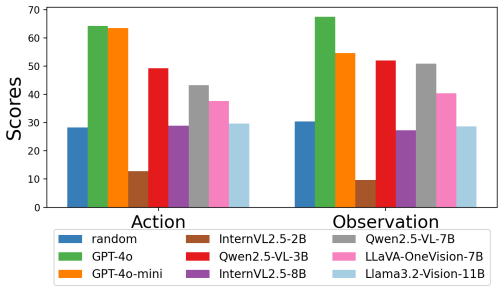


Figure 7: Model performance on Local Observation-Action Reasoning.

**Evaluation Metrics** Our benchmark includes both multiple-choice reasoning and embodied navigation execution tasks. For multiple-choice questions, we follow standard practice [5] and use *Accuracy* as the primary metric, which measures whether the model selects the correct answer from a set of candidates based on the provided information. For execution tasks, we adopt standard metrics in embodied navigation [9, 30]. *Success Rate (SR)* measures the percentage of episodes where the target object is visible from the agent's final viewpoint, defined as being within a 3-meter radius. *Success weighted by Path Length (SPL)* adjusts SR by path efficiency and is computed as:

$$\text{SPL} = \frac{1}{N}\sum_{i=1}^{N} S_i \cdot \frac{\ell_i}{\max(\ell_i, p_i)}. \tag{6}$$

where $N$ is the number of episodes, $S_i \in \{0, 1\}$ indicates success, $\ell_i$ is the shortest path, and $p_i$ is the path length.

**VLN-Bench (tiny) Human Performance** To provide an upper-bound reference, we additionally report human performance on a compact subset of VLN-Bench, denoted as VLN-Bench (tiny), which was manually annotated and evaluated following the same protocol.

### 5.2 Performance

We begin by examining the relationship between comprehension and execution. As shown in Table 1, model performance on comprehension and execution tasks remains closely aligned. Among closed models, o4-mini achieves the highest comprehension average (59.66%) and maintains competitive execution performance (28.98%). GPT-4o follows with 53.34% and 41.33%, respectively, suggesting

Table 2: Impact of map information on GPT-4o.

| Diff. | Map | SR | SPL | Avg | Gain |
|---|---|---|---|---|---|
| Easy | ✗ | 67.36 | 54.31 | 60.84 | – |
| | ✓ | 70.14 | 54.11 | 62.13 | +1.29 |
| Med. | ✗ | 41.67 | 35.71 | 38.69 | – |
| | ✓ | 46.53 | 39.86 | 43.20 | +4.51 |
| Hard | ✗ | 27.78 | 21.15 | 24.47 | – |
| | ✓ | 29.17 | 22.32 | 25.75 | +1.28 |


Figure 8: Distribution of navigation error types identified from manual analysis.

that o4-mini excels in understanding navigation instructions, while GPT-4o is stronger in executing them. Among open-source models, Qwen2.5-VL-7B achieves the best overall performance (45.26%, 21.77%), approaching GPT-4o-mini (46.42%, 27.99%) and demonstrating potential for practical deployment in real-world robotics. Turning to comprehension subtasks, InternVL2.5-2B performs strongly on Global Instruction Alignment (67.25%), even surpassing GPT-4o (51.33%). However, its accuracy drops sharply on more challenging reasoning tasks. In particular, Progress Estimation remains a consistent weakness across models; aside from GPT-4o (42.90%), all others perform poorly, highlighting current MLLMs' limitations in temporal reasoning. We next analyze how models perform across navigation difficulty levels. Most open-source models can only reliably complete Easy episodes, while GPT-4o maintains relatively strong results across all levels, suggesting better generalization. These findings suggest several overarching insights. First, comprehension and execution abilities are strongly linked. Second, temporal reasoning, particularly progress tracking, remains a major bottleneck. Third, compact open-source models like Qwen2.5-VL-7B can offer competitive performance with significantly lower resource requirements, making them promising for embodied applications.

### 5.3 Discussion

**Breakdown of Distractor Types in Instruction Alignment** We further analyze performance on the Global Instruction Alignment task by breaking down results across four distractor types: *basic*, *direction*, *object*, and *shuffle*. As shown in Figure 6, most models handle the basic condition well, indicating their ability to reject unrelated instructions. However, performance under *direction* and *object* perturbations varies significantly across models, suggesting inconsistent grounding of spatial terms and landmarks. Notably, all models perform poorly under the *shuffle* condition, where sub-instructions are reordered but their content remains unchanged. This result is particularly revealing: despite the presence of the same entities and actions, altering the temporal structure makes the instruction much harder for models to interpret. The models' failure in this setting highlights their limited ability to reason about temporal order within complex instructions. This finding aligns with the low scores observed in the Progress Estimation task, reinforcing that current MLLMs struggle with temporal understanding across both instruction-level and trajectory-level reasoning.

**Future-Action and Future-Observation Reasoning** We analyze performance on the Local Observation-Action Reasoning task, which includes two subtasks: Future-Action and Future-Observation Prediction. As shown in Figure 7, models show consistent performance across both, with GPT-4o clearly outperforming all others, consistent with its strong results in Navigation Execution. These subtasks reflect complementary reasoning skills. Future-Action Prediction tests whether a model can infer the spatial transition between two views, while Future-Observation Prediction requires anticipating how the environment changes after a given action. Both capabilities are critical for navigation, where agents should reason about cause and effect in spatial transitions.

**Effect of Map Information on Action Decisions** Although our benchmark evaluations assume no access to map information, reflecting real-world constraints, we investigate whether providing map connectivity can enhance action selection. Specifically, we follow the approach introduced in MapGPT, where topological relationships between explored nodes are encoded as text prompts. Using GPT-4o, we compare performance with and without map input across different difficulty levels. As shown in Table 2, the presence of map information consistently improves success rates, with the largest gain observed under medium difficulty, yielding an increase of 4.86 percentage points. This suggests that access to structured spatial context can facilitate better high-level reasoning and planning, especially in medium complexity settings where spatial ambiguity is more common.

Table 3: Performance comparison with and without CoT prompting.

| Model | Navigation Comprehension | | | | Navigation Execution | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Global | Progress | Local | Comp. Avg | Easy | | Medium | | Hard | | Exec. Avg |
| | Accuracy | | | | SR | SPL | SR | SPL | SR | SPL | |
| GPT-4o | 51.33 | 42.90 | 65.80 | 53.34 | 67.36 | 54.31 | 41.67 | 35.71 | 27.78 | 21.15 | 41.33 |
| GPT-4o + CoT | 60.42 | 40.20 | 60.75 | 53.79 | 61.11 | 49.04 | 44.44 | 36.88 | 30.56 | 23.20 | 40.87 |
| Qwen2.5-VL-7B | 57.58 | 31.20 | 47.00 | 45.26 | 41.67 | 32.55 | 22.92 | 17.43 | 10.42 | 5.67 | 21.77 |
| Qwen2.5-VL-7B + CoT | 60.50 | 31.40 | 48.00 | 46.63 | 43.75 | 33.06 | 22.92 | 15.62 | 11.11 | 7.19 | 22.28 |

**Effect of Chain-of-Thought** We incorporated Chain-of-Thought (CoT) prompting following [66] by prepending "Let's think step by step" to the instruction input. As shown in Table 3, experiments were conducted using two of the strongest models in our benchmark: GPT-4o and Qwen2.5-VL-7B. The results show that CoT prompting brings noticeable improvement in the Global Instruction Alignment task. GPT-4o improved by 9.09%, and Qwen2.5-VL-7B improved by 2.92%. However, the gains in other comprehension tasks were marginal or slightly negative. We hypothesize that simple CoT prompting does not sufficiently enhance performance in spatial or temporal reasoning tasks, which often require more structured, multi-step planning rather than generic step-by-step thinking. For the navigation execution task, we observed little benefit from CoT prompting. This is likely because the task itself already follows a step-by-step process: at each time step, the model receives the full instruction history and must decide the next action. Therefore, additional CoT prompting provides limited benefit in this context.

**Error Analysis** (1) We manually analyze 100 failed cases to understand model failures. Based on thought traces and action sequences, we identify four common error types: (a) *Incorrect Plan*: the plan misaligns with the instruction; (b) *Misaligned Action*: the plan is valid, but the chosen movement does not follow it; (c) *Failure to Stop*: the agent overshoots the goal or stops early; and (d) *Hallucinated Movement*: the model selects a nonexistent location. The error distribution is shown in Figure 8. These patterns align with weaknesses in comprehension tasks. For example, type (c) reflects poor *Progress Estimation*. This suggests execution failures often stem from temporal and spatial reasoning limitations, reinforcing the diagnostic value of NavBench.

(2) We further examine the impact of trajectory length on temporal reasoning. Test samples are grouped by length into short (1–2 steps), medium (3–4), and long (5+). For GPT-4o, the error rate increases from 35.3% (short) to 42.9% (medium) and 76.1% (long), showing that longer trajectories amplify temporal reasoning challenges. In contrast, weaker models such as LLaVA-OneVision-7B and InternVL2.5-2B maintain high error rates across all lengths, indicating persistent difficulty in progress estimation regardless of path complexity.

**Real-World Validation** To assess the feasibility of our real-world deployment pipeline, we conduct a pilot study in an indoor environment using GPT-4o and Qwen2.5-VL-7B, the top proprietary and open-source models from our benchmark. Each model is tested on 10 cases, achieving success rates of 60% and 40%, respectively. These results show that both can handle simple navigation tasks in real-world settings. Their success trends mirror execution performance in Table 1, where both models outperform others in their categories. This suggests that NavBench's simulation-based evaluation reliably reflects real-world embodied performance.

## 6 Conclusion

This paper presents NavBench, a diagnostic benchmark designed to evaluate MLLMs in embodied navigation under zero-shot settings. It decomposes the evaluation into two components: Navigation Comprehension, which evaluates global instruction alignment, temporal progress estimation, and local observation-action reasoning through three cognitively grounded tasks, and Navigation Execution, which examines step-by-step decision-making across varying levels of difficulty. Additionally, we develop a pipeline for real-world deployment of MLLM-driven agents. Through evaluation and targeted analysis, NavBench reveals limitations in temporal understanding and action grounding that are not captured by standard success metrics. It also shows that lightweight open-source models can be effective in simpler navigation scenarios. We hope NavBench can serve as a useful resource for analyzing the embodied capabilities of MLLMs and supporting future work in this direction.

## References


[1] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[2] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[3] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[4] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017.

[5] Chaoyou Fu, Yuhan Dai, Yondong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Rongrong Ji, and Xing Sun. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *ArXiv*, abs/2405.21075, 2024.

[6] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.

[7] Wenxiao Cai et al. Spatialbot: Precise spatial understanding with vision language models. In *IEEE international conference on robotics and automation*, 2025.

[8] Jihan Yang, Shusheng Yang, Anjali Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in Space: How Multimodal Large Language Models See, Remember and Recall Spaces. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[9] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian D. 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.

[10] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33:4247–4258, 2020.

[11] Benjamin Kuipers. The spatial semantic hierarchy. *Artificial intelligence*, 119(1-2):191–233, 2000.

[12] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Zhong Muyan, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Intern vl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2023.

[13] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. *ArXiv*, abs/2204.14198, 2022.

[14] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *ArXiv*, abs/2308.12966, 2023.

[15] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26286–26296, 2023.

[16] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019.

[17] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 3195–3204, 2019.

[18] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326, 2019.

[19] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*, 2023.

[20] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European Conference on Computer Vision*, pages 216–233. Springer, 2025.

[21] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023.

[22] Chenming Zhu, Tai Wang, Wenwei Zhang, Kai Chen, and Xihui Liu. Scanreason: Empowering 3d visual grounding with reasoning capabilities. In *European Conference on Computer Vision*, 2024.

[23] 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 9979–9988, 2020.

[24] Zehao Wang, Minye Wu, Yixin Cao, Yubo Ma, Meiqi Chen, and Tinne Tuytelaars. Navigating the nuances: A fine-grained evaluation of vision-language navigation. *ArXiv*, abs/2409.17313, 2024.

[25] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *IEEE international conference on robotics and automation*, pages 3357–3364. IEEE, 2017.

[26] Jacob Krantz, Stefan Lee, Jitendra Malik, Dhruv Batra, and Devendra Singh Chaplot. Instance-specific image goal navigation: Training embodied agents to find object instances. *arXiv preprint arXiv:2211.15876*, 2022.

[27] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9339–9347, 2019.

[28] Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. In *Advances in Neural Information Processing Systems*, 2021.

[29] Yue Zhang et al. Vision-and-language navigation today and tomorrow: A survey in the era of foundation models. *Transactions on Machine Learning Research*, 2024.

[30] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 4392–4412, 2020.

[31] Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. Vision-and-dialog navigation. In *CoRL*, pages 394–406, 2019.

[32] Hongcheng Wang, Andy Guan Hong Chen, Xiaoqi Li, Mingdong Wu, and Hao Dong. Find what you want: Learning demand-conditioned object attribute space for demand-driven navigation. In *Advances in Neural Information Processing Systems*, 2023.

[33] Yuankai Qi, Zizheng Pan, Shengping Zhang, Anton van den Hengel, and Qi Wu. Object-and-action aware model for visual language navigation. In *European Conference on Computer Vision*, pages 303–317. Springer, 2020.

[34] Yicong Hong, Cristian Rodriguez, Yuankai Qi, Qi Wu, and Stephen Gould. Language and visual entity relationship graph for agent navigation. *Advances in Neural Information Processing Systems*, 33:7685–7696, 2020.

[35] Dong An, Yuankai Qi, Yan Huang, Qi Wu, Liang Wang, and Tieniu Tan. Neighbor-view enhanced model for vision and language navigation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 5101–5109, 2021.

[36] Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao. Towards learning a generic agent for vision-and-language navigation via pre-training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13137–13146, 2020.

[37] Pierre-Louis Guhur, Makarand Tapaswi, Shizhe Chen, Ivan Laptev, and Cordelia Schmid. Airbert: In-domain pretraining for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1634–1643, 2021.

[38] Arjun Majumdar, Ayush Shrivastava, Stefan Lee, Peter Anderson, Devi Parikh, and Dhruv Batra. Improving vision-and-language navigation with image-text pairs from the web. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pages 259–274. Springer, 2020.

[39] Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. *Advances in neural information processing systems*, 34:5834–5847, 2021.

[40] Yanyuan Qiao, Yuankai Qi, Yicong Hong, Zheng Yu, Peng Wang, and Qi Wu. Hop: History-and-order aware pre-training for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15418–15427, 2022.

[41] Yanyuan Qiao, Yuankai Qi, Yicong Hong, Zheng Yu, Peng Wang, and Qi Wu. Hop+: History-enhanced and order-aware pre-training for vision-and-language navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7):8524–8537, 2023.

[42] Dong An, Yuankai Qi, Yangguang Li, Yan Huang, Liang Wang, Tieniu Tan, and Jing Shao. Bevbert: Multimodal map pre-training for language-guided navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2737–2748, 2023.

[43] Rui Liu, Xiaohan Wang, Wenguan Wang, and Yi Yang. Bird's-eye-view scene graph for vision-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10968–10980, 2023.

[44] Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. Gridmm: Grid memory map for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15625–15636, 2023.

[45] Zun Wang, Jialu Li, Yicong Hong, Yi Wang, Qi Wu, Mohit Bansal, Stephen Gould, Hao Tan, and Yu Qiao. Scaling data generation in vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12009–12020, 2023.

[46] Aishwarya Kamath, Peter Anderson, Su Wang, Jing Yu Koh, Alexander Ku, Austin Waters, Yinfei Yang, Jason Baldridge, and Zarana Parekh. A new path: Scaling vision-and-language navigation with synthetic instructions and imitation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10813–10823, 2023.

[47] Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. Navid: Video-based vlm plans the next step for vision-and-language navigation. *arXiv preprint arXiv:2402.15852*, 2024.

[48] Kaiwen Zhou, Kaizhi Zheng, Connor Pryor, Yilin Shen, Hongxia Jin, Lise Getoor, and Xin Eric Wang. Esc: Exploration with soft commonsense constraints for zero-shot object navigation. In *International Conference on Machine Learning*, pages 42829–42842. PMLR, 2023.

[49] Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23171–23181, 2023.

[50] Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. Vlfm: Vision-language frontier maps for zero-shot semantic navigation. In *IEEE international conference on robotics and automation*, pages 42–48. IEEE, 2024.

[51] Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. *arXiv preprint arXiv:2406.04882*, 2024.

[52] Gengze Zhou, Yicong Hong, and Qi 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.

[53] Yanyuan Qiao, Wenqi Lyu, Hui Wang, Zixu Wang, Zerui Li, Yuan Zhang, Mingkui Tan, and Qi Wu. Open-nav: Exploring zero-shot vision-and-language navigation in continuous environment with open-source llms. *arXiv preprint arXiv:2409.18794*, 2024.

[54] Jiaqi Chen, Bingqian Lin, Ran Xu, Zhenhua Chai, Xiaodan Liang, and Kwan-Yee Wong. Mapgpt: Map-guided prompting with adaptive path planning for vision-and-language navigation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9796–9810, 2024.

[55] Angel X. Chang, Angela Dai, Thomas A. Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from RGB-D data in indoor environments. In *3DV*, pages 667–676, 2017.

[56] Yibo Cui, Liang Xie, Yakun Zhang, Meishan Zhang, Ye Yan, and Erwei Yin. Grounded entity-landmark adaptive pre-training for vision-and-language navigation. *IEEE/CVF International Conference on Computer Vision*, pages 12009–12019, 2023.

[57] Yicong Hong, Cristian Rodriguez, Qi Wu, and Stephen Gould. Sub-instruction aware vision-and-language navigation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 3360–3376, 2020.

[58] Xinyu Wang, Bohan Zhuang, and Qi Wu. Are large vision language models good game players? In *Proceedings of the International Conference on Learning Representations*, 2025.

[59] Bosi Wen, Pei Ke, Xiaotao Gu, Lindong Wu, Hao Huang, Jinfeng Zhou, Wenchuang Li, Binxin Hu, Wendy Gao, Jiaxin Xu, Yiming Liu, Jie Tang, Hongning Wang, and Minlie Huang. Benchmarking complex instruction-following with multiple constraints composition. *ArXiv*, abs/2407.03978, 2024.

[60] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yiming Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, Conghui He, Botian Shi, Xingcheng Zhang, Han Lv, Yi Wang, Wenqi Shao, Pei Chu, Zhongying Tu, Tong He, Zhiyong Wu, Hui Deng, Jiaye Ge, Kaiming Chen, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahu Lin, Yunfeng Qiao, Jifeng Dai, and Wenhai Wang. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *ArXiv*, abs/2412.05271, 2024.

[61] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *ArXiv*, abs/2502.13923, 2025.

[62] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[63] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[64] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[65] LMDeploy Contributors. Lmdeploy: A toolkit for compressing, deploying, and serving llm. `https://github.com/InternLM/lmdeploy`, 2023.

[66] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Annual Conference on Neural Information Processing Systems*, 2022.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the contributions of the paper, including the design of NavBench, task decomposition, difficulty stratification, and real-world deployment, all of which are substantiated by experiments. (See Section 1 and Abstract)

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations of the proposed benchmark in the supplementary material.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include any theoretical results or formal proofs; it is purely empirical and benchmarking-based.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper includes sufficient implementation details and describes benchmark construction, task setup, and evaluation procedures. (See Sections 3–5)

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will release the dataset and code. Anonymized supplementary material includes reproduction instructions. (Details in Supplementary)

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper details all relevant settings, including model types, deployment methods, simulator configuration, robot setup, and evaluation metrics. (See Section 5.1)

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We do not report error bars or significance tests. The evaluation focuses on average performance across tasks and difficulty levels.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We describe the hardware setup (e.g., A6000 GPU, robot platform), and simulator environments used for training and testing. (See Section 5.1)

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms with the NeurIPS Code of Ethics. All data used are from publicly available benchmarks.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This work presents a benchmark for evaluating MLLMs in embodied navigation using publicly available data in simulated and controlled lab environments. It does not involve human subjects, private data, or direct deployment scenarios, and we do not foresee any immediate societal impact.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not release any models or datasets that pose a high risk for misuse. All evaluated models are publicly available, and the benchmark is constructed from existing datasets with no sensitive content.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All used datasets (e.g., R2R, RxR, GEL-R2R) and models (e.g., GPT-4o, Qwen2.5-VL) are publicly available and properly cited in the paper. We respect their terms of use and follow the licenses specified by their original creators.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper introduces a new benchmark, NavBench, constructed from existing datasets but reorganized into new evaluation tasks with structured question-answer pairs and difficulty annotations. We provide documentation and plan to release anonymized assets upon acceptance.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve any new crowdsourcing or research with human participants. All data used were sourced from publicly available datasets that already contain necessary annotations.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve research with human subjects as defined by IRB standards. All human judgments were conducted internally by the authors without collecting personal or sensitive data.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: This work systematically evaluates the embodied reasoning capabilities of existing LLMs (e.g., GPT-4o, Qwen2.5-VL) in navigation tasks. These models are not developed by the authors, but they are central to the paper's evaluation design and analysis. Their usage is described in detail throughout the paper.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.