CVPR
#2124

CapNav: Benchmarking Vision Language Models on Capability-conditioned Indoor Navigation

Anonymous CVPR submission

Paper ID 2124

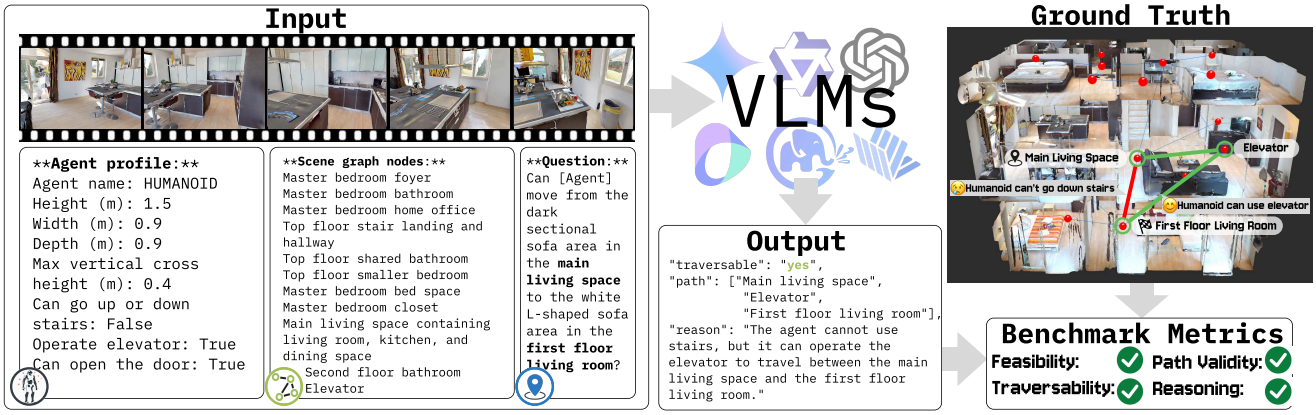

Figure 1. We introduce Capability-Conditioned Navigation (***CapNav***), a benchmark designed to evaluate how well VLMs can navigate complex indoor spaces given an agent's specific physical and operational capabilities. CapNav inputs (1) a tour video of an indoor space, (2) nodes of its navigation graph, (3) an agent's mobility profile, and (4) a navigation task, and evaluates VLM outputs in task feasibility, path validity, route traversability, and reasoning validity.

Abstract

*Vision-Language Models (VLMs) have shown remarkable progress in Vision-Language Navigation (VLN), offering new possibilities for navigation decision-making that could benefit both robotic platforms and human users. However, real-world navigation is inherently conditioned by the agent's mobility constraints. For example, a sweeping robot cannot traverse stairs while a quadruped can. We introduce Capability-Conditioned Navigation (**CapNav**), a benchmark designed to evaluate how well VLMs can navigate complex indoor spaces given an agent's specific physical and operational capabilities. CapNav defines five representative human and robot agents, each described with physical dimensions, mobility capabilities, and environmental interaction abilities. CapNav provides 45 real-world indoor scenes, 473 navigation tasks, and 2365 QA pairs to test if VLMs can traverse indoor environments based on agent capabilities. We evaluate 13 modern VLMs and find that current VLM's navigation performance drops sharply as mobility constraints tighten, and that even state-of-the-art models struggle with obstacle types that require reasoning on spatial dimensions. We close by discussing the implications for capability-aware navigation and the opportunities for advancing embodied spatial reasoning in future VLMs.*

1. Introduction

As vision–language models (VLMs) advance in visual grounding [43] and spatial reasoning [9], they have been increasingly applied to navigation tasks—helping people reach destinations [14, 28] and robots in planning and executing movement decisions [17]. Numerous frameworks [1, 14, 18, 22, 58] and benchmarks [12, 45, 52, 57] have been proposed to evaluate these navigation capabilities, and VLMs are now commonly used as drop-in "navigation assistants" or planning modules for robots and assistive systems [14, 42, 60]. However, real-world navigation is inherently conditioned by the agents' movement capabilities.

CVPR
#2124

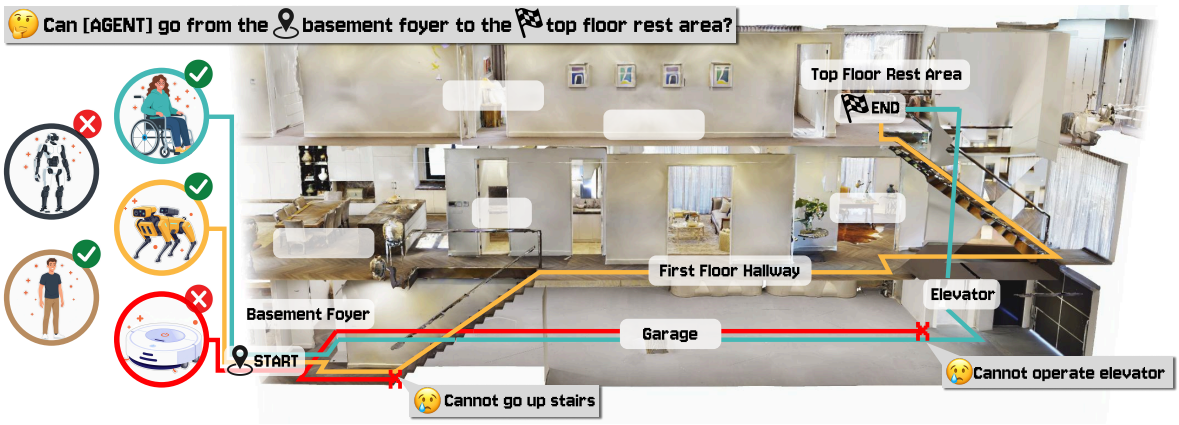

Figure 2. The CapNav benchmark evaluates whether VLMs can correctly ground differences in agent mobility capabilities when generating navigation plans. This example demonstrates a navigation task that has different feasibility and path for different agents.

Thus, a critical gap remains: existing evaluations rarely test whether VLMs can reason about an agent's physical capabilities or align them with environmental constraints.

Consider the multi-story building shown in Figure 2. To reach the top-floor rest area, VLM should route the wheelchair user via the elevator, but the quadruped robot via the stairwell (since it cannot operate the elevator). Similarly, a non-disabled human may squeeze through a cluttered corridor while a wide humanoid robot cannot. Without capability-aware evaluation, VLM-based navigation may recommend actions that are infeasible or even unsafe for a given agent [15, 29].

We introduce *CapNav*, a benchmark for evaluating VLM's navigation performance across five representative agent types: adults with no motor disabilities, wheelchair users, sweeping robots, humanoid robots, and quadrupedal robots. CapNav transforms HM3D [38] and MP3D [6] scenes into walkthrough videos, each with realistic mobility obstacles. Following the graph abstraction popularized by R2R and widely adopted in Habitat-based evaluations [1, 23, 40], we annotate each scene as a navigation graph, where nodes denote sub-spaces and edges mark per-agent traversability. In total, CapNav contains 45 scenes, 2365 navigation tasks, and 5075 traversability annotations as ground truth. Among the 2K+ navigation tasks, four major types of obstacles are observed: stairs (N=520), door sill/floor height difference (82), narrow or cluttered pathway (438), and lack of turning space (28).

At inference, we input: (1) a tour video of the indoor space (*Avg length*=160.38 secs), (2) nodes of the navigation graph, (3) the target agent's mobility profile, (4) and a navigation task (*e.g.,* Can [Agent] go from the basement foyer to the top floor rest area?). The VLM responds with navigation feasibility (binary), the navigational graph path, and its reasoning. We benchmark 13 state-of-the-art VLMs across four axes: **feasibility classification** if a task is feasible for an agent; **path validity** if the predicted graph path exists; **route traversability** if the agent can actually traverse the predicted path; and **reasoning validity** if the provided reasoning matches ground truth annotations. We report an aggregate *CapNav Score* combining these components and break down results per agent type, per obstacle type, and per model parameters.

Across models, we observe strong overall navigation performance in high-mobility settings but degradation of performance as mobility constraints tighten: performance is best under human-like assumptions, and worst for the most limited agent type. We also find performance sensitivity to input frame rate, thinking settings, and model size. Additionally, we notice that navigation challenges that require reasoning on spatial dimension is the most challenging and very poorly addressed. These trends highlight the need for capability-aware modeling and training, as well as for explicit checks against potential hallucination [15, 29].

Our contributions are threefold:

1. **The CapNav benchmark.** We introduce CapNav, a capability-conditioned VLN benchmark that evaluates how VLMs navigate complex indoor environments under realistic mobility constraints across five representative human and robot embodiments.
2. **Comprehensive evaluation of VLMs.** We conduct a head-to-head assessment of 13 state-of-the-art VLMs on capability-aware feasibility prediction, path validity, route traversability, and reasoning quality.
3. **Guidelines and resources.** We analyze the effects of input frame rates, model settings, obstacle types, and failure modes, providing actionable guidelines for capability-aware navigation. We release the full dataset—videos, tasks, agent profiles, and 5k+ traversability annotations—along with an interactive annotation interface for extending CapNav to new agents and environments.

2. Related Work

2.1. Vision Language Navigation and Benchmarks

Vision-Language Navigation (VLN) is the task of interpreting natural language instructions and grounded visual observations to move through an environment and reach a goal [1, 61]. Since its initial introduction in the Room-to-Room (R2R) dataset [1], which enables step-by-step movement control in a sparse indoor navigation graph, numerous variants have emerged. For example, Room-for-Room (R4R) [19] and Room-Across-Room (RxR) [24] evaluate longer paths, linguistic diversity, and middle steps; TOUCHDOWN [7], StreetLearn [33], CityLearn [5], CityWalker [31] and Talk2Nav [49] expanded to city environments. Many works, like REVERIE [37], ALFRED [44], ObjectNav [2], and GOAT-bench [21] also combine navigation with object grounding or interaction.

Recently, as VLMs demonstrate strong spatial reasoning capabilities, the input of VLN has shifted from stepwise egocentric exploration to video-based global observation. Instead of incrementally perceiving local viewpoints, many recent works provide models with a pre-collected video or panoramic scan covering the entire environment. This paradigm originated from attempts to decouple high-level reasoning from low-level control. For instance, VideoNavQA [3] replaced interactive navigation with an oracle trajectory video to isolate visual reasoning performance in embodied QA. Later, OpenEQA [32] explicitly adopted a passive, video-based setting to evaluate models' spatial memory and reasoning without the confounding factor of exploration errors. These works demonstrate that foundation VLMs excel when given global context rather than partial observations.

Our evaluation goal for capability-conditioned navigation benefits from full-space video observation, since it allows VLMs to reason about traversability and route feasibility across the entire scene. At the same time, the graph-based abstraction from classical VLN [1, 24] enables structured comparisons across agents and mobility constraints. CapNav combines global video-based context with the sparse graph representation to allow systematic evaluation of VLMs' capability-conditioned spatial reasoning in complex environments and across diverse embodiments.

2.2. Navigation with mobility constraints

Past navigation work had considered a wide range of specific mobility constraints, like the mobility of wheelchair users [13, 39, 59], people who are blind or low vision [11, 34, 50], quadrupeds [16, 25, 30], humanoids [8], and also vehicles [20, 27]. Many works even consider variant mobility settings that co-exist, either in simulators [53–55], or preference discovery [26]. These studies demonstrate that mobility constraints significantly affect feasible routes and navigation strategies.

However, existing work focuses on fragmented embodiments and fails to provide comparison of navigation performance across multiple embodiments on the same task. This gap is starting to be addressed in the latest work: NaviTrace [51] benchmarks single-image navigation under different embodiments, and VAMOS [4] train an affordance model to encourage embodiment-specific navigation. But so far, no work has benchmarked VLMs for cross-embodiment navigation in complex indoor environments.

3. The CapNav Benchmark

We introduce *Capability-conditioned Navigation* (CapNav), a benchmark for assessing how well can VLMs navigate in built environments given particular physical capabilities.

3.1. Problem Statement

In the CapNav task, each query instance is defined by a *Space–Task–Capability* triple

$$\langle \mathcal{S}, \tau, \mathbf{a} \rangle,$$

where the space $\mathcal{S}$ is represented by a touring video and a set of key spatial nodes; τ is a natural-language navigation goal specifying the source and target locations; and $\mathbf{a}$ is an agent profile encoding physical dimensions and operational abilities. Given $\langle \mathcal{S}, \tau, \mathbf{a} \rangle$, a vision language model produces

$$(\hat{y}, \hat{P}, \hat{\rho}) \;=\; f_\theta(\mathcal{S}, \tau, \mathbf{a}),$$

where $\hat{y} \in \{0, 1\}$ denotes task feasibility (CAN / CANNOT complete), $\hat{P} = [v_0, \ldots, v_m]$ is a node sequence representing the proposed navigation path, and $\hat{\rho}$ is a concise rationale explaining infeasibility when $\hat{y} = 0$.

This formulation enables CapNav to holistically evaluate a VLM's capability-aware navigation ability. We assess performance from four complementary aspects: (i) *navigation feasibility*, (ii) *path validity*, (iii) *traversability accuracy*, and (iv) *reasoning quality* for infeasible cases.

Space. Each space $\mathcal{S}$ is a scanned 3D indoor environment comprising multiple interconnected rooms. As discussed in subsection 2.1, we provide spatial information as a rendered touring video $\mathcal{X} = \{x_t\}_{t=1}^{T}$. To better ground the navigation to language description, we manually create a connectivity graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ over semantically meaningful spatial nodes (e.g., *kitchen*, *entry foyer*, *hallway*) and edges that denote directly walkable connections between the nodes. At inference, we only provide the video and the node list.

Task. A navigation task τ is a natural-language instruction specifying a movement goal from a source node to a

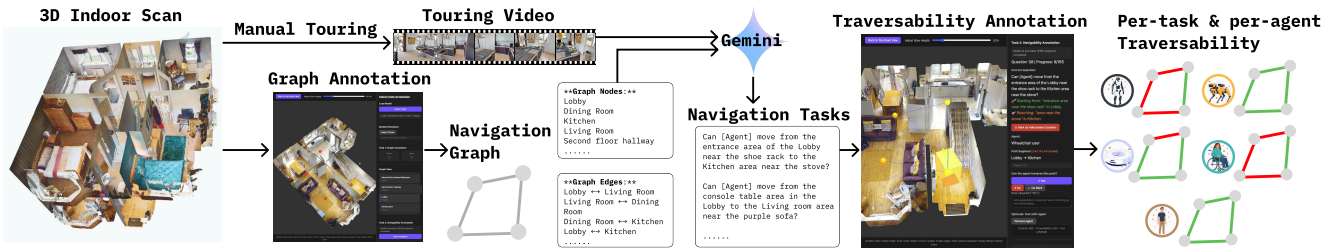

Figure 3. Overview of CapNav's data construction: Starting from a 3D indoor scan, we manually record a touring video and a navigation graph. We then use Gemini to generate natural language navigation tasks. Finally, per-task and per-agent traversability are annotated by manually controlling agents in the annotation interface.

target node (*e.g.,* "*From the wooden cabinet in the entrance foyer, go to the desk area in the master bedroom.*"). Formally, each goal τ can be expressed as a directed pair:

$$\tau : (v_{\text{src}}, d_{\text{src}}) \rightarrow (v_{\text{tgt}}, d_{\text{tgt}}), \quad v_{\text{src}}, v_{\text{tgt}} \in \mathcal{V},$$

where v_{src} and v_{tgt} denote the source and target nodes in $\mathcal{V}$, and d_{src} and d_{tgt} are their corresponding natural-language descriptions specifying the precise starting and ending positions within those nodes (*e.g.,* "*at the wooden cabinet in the entrance foyer*", "*beside the desk area in the master bedroom*"). This representation captures both the high-level spatial nodes and the finer-grained textual localization needed for realistic indoor navigation reasoning.

Agent Profiles We specify the mobility capabilities $\mathbf{a}$ with five distinct but representative agent profiles that cover human mobility and common robot platforms. **Adult with no motor disabilities** represent a default capability condition where all indoor routes can be achieved; **Wheelchair users** cannot go up stairs and require enough clearance for passing and turning; **Humanoid robot** cannot go up/down stairs and require clearance for passing; **Sweeping robots** require flat floor surface; **Quadrupedal robots** can traverse most indoor spaces but cannot operate doors/elevator.

Each profile is described by a capability json $\mathbf{a} = (\phi, \kappa, \mu)$, where ϕ captures the physical footprint; κ captures vertical traversal limits, including vertical height of a single obstacle, and whether the agent can cross continuous stairs; and μ captures operation/manipulation abilities. See Figure 4 for examples. These attributes directly affect traversability on edges $\mathcal{E}$ and on overall tasks τ.

3.2. Ground Truth of Traversability

Traversability is manually annotated per-task and per-agent type. They are binary labels at the *edge* level, allowing multiple possible routes and providing more precise reasoning if a route is not traversable. For each navigation task τ and embodiment $\mathbf{a}$, annotators iterate each possible simple path that connects v_{src} and v_{tgt} in $\mathcal{G}$, and assign $g_e^{(\mathbf{a})} \in \{0, 1\}$,

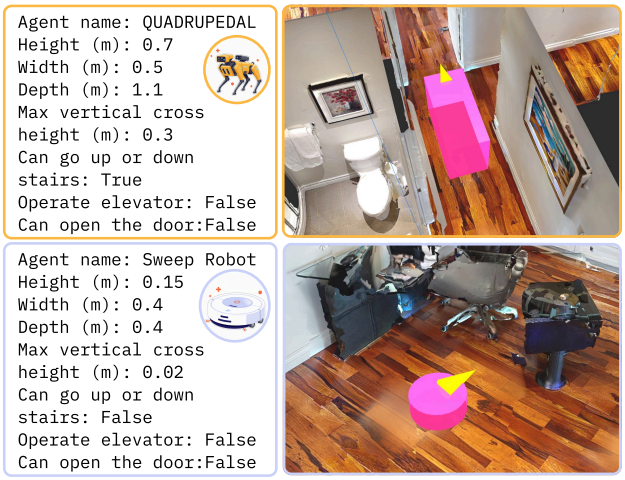

Figure 4. Examples of CapNav's agent profiles. Left: each profile specifies the physical dimensions and functional capabilities. Right: the agents' physical dimensions are rendered and maneuvered in the 3D scenes to help confirm traversability.

to each edge e on these paths, indicating whether e is passable under $\mathbf{a}$ given local spatial geometry and mobility capabilities. Reasons for non-traversability are recorded as text descriptions provided by the annotator (*e.g.,* "*cannot go up/down stairs*"). The annotation UI visualizes 3D colliders matching ϕ and supports manually controlled moving/turning to verify clearance and turning space (Fig. 3).

Given per-edge labels, we can then define the feasibility ground truth for a task (s, g) as, if there exists at least one simple path (an acyclic sequence of distinct nodes) $P(v_{\text{src}}, v_{\text{tgt}})$ that connects v_{src} and v_{tgt} in $\mathcal{G}$, where all edges are traversable:

$$y^\star = \mathbb{I}\big[\exists\, P(v_{\text{src}}, v_{\text{tgt}}) : \forall (u, v) \in P,\ g_{(u,v)}^{(\mathbf{a})} = 1\big],$$

3.3. Dataset

To establish a benchmark for the CapNav task, we collected a dataset of 3D indoor scenes from HM3D [38] and Matterport3D [6]. As shown in Figure 3, we record touring videos

by manually moving through each space in the Habitat simulator [40]. We rendered video at 2FPS and from human-eye height ($1.5m$) with a field of view of 75, in order to mimic casual hand-held walkthroughs. We then construct navigation graphs for the entire indoor scene using the annotation interface. Nodes $v \in \mathcal{V}$ are annotated with semantic labels $c(v)$ and approximate 3D positions, while edges $e = (u, v) \in \mathcal{E}$ are manually added to denote bidirectional, directly walkable connections. We provide the space information as a video and a space node list to Gemini 2.5 Pro, and prompt it to generate navigation tasks in such space. We then manually check the validity of the task. For valid tasks, we manually annotate edge-level traversability for all five embodiments using our custom annotation interface and write down the reasons for non-traversable cases. We exclude scenes with major holes, disconnected subspaces, repetitive semantics, or trivially small layouts—resulting in 45 indoor scenes, each rendered at 2 FPS with average video length $160.38s$. On average, each indoor scene has 13.8 nodes, 14.5 edges. In total, there are 2365 navigation tasks, 5075 traversability labels, among which 3945 are positive and 1130 are negative.

3.4. Metrics & Scoring

We evaluate model outputs against ground truth using four complementary metrics, reported both overall and per-embodiment. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the space graph, let $P(v_{\text{src}}, v_{\text{tgt}})$ denote a simple path in $\mathcal{G}$, and let $E(\hat{P}) \subseteq \mathcal{E}$ be the ordered edge set of a predicted path $\hat{P}$.

(1) Feasibility Classification (Feas-F1). Let $\hat{y} \in \{0, 1\}$ be the model's feasibility prediction and $y^\star$ the ground truth (Sec. 3.2). We report positive-class precision, recall, and F_1:

(2) Path Validity (PV). A predicted route $\hat{P} = [v_0, \ldots, v_m]$ is considered valid if (i) $v_i \in \mathcal{V}$ for all i, (ii) $(v_i, v_{i+1}) \in \mathcal{E}$ for all i, (iii) $v_0 = v_{\text{src}}$ and $v_m = v_{\text{tgt}}$, and (iv) $\hat{P}$ is a simple path (no node repetitions). We report:

$$\text{PV} = \mathbb{E}\big[\mathbb{I}(\hat{P} \in \mathcal{P}_{\text{simple}}(v_{\text{src}}, v_{\text{tgt}}))\big].$$

(3) Route Traversability Accuracy (RTA). Conditioned on $\hat{y} = 1$ and a valid $\hat{P}$, we evaluate the fraction of edges in $\hat{P}$ that are actually traversable under embodiment $\mathbf{a}$:

$$\text{RTA}(\hat{P}, \mathbf{a}) = \frac{\sum_{e \in E(\hat{P})} g_e^{(\mathbf{a})}}{|E(\hat{P})|},$$

where $g_e^{(\mathbf{a})} \in \{0, 1\}$ is the edge-level traversability label. All edges are assigned unit weight so each edge contributes equally. RTA $= 1$ indicates a fully valid route, while lower values quantify partial failures. This metric serves a role analogous to path-quality measures such as SPL [1], but conditions on embodiment-specific traversability.

(4) Reasoning Validity (RV). For infeasible predictions ($\hat{y} = 0$), the model must (i) return a proposed route $\hat{P}$ and (ii) provide a short rationale $\hat{\rho}$ explaining the failure. Let $\mathcal{R}^\star(\hat{P}, \mathbf{a})$ be the set of annotated failure reasons, we define an LLM-as-judge function J_{LLM} that returns 1 if the model's rationale semantically matches annotation and 0 otherwise. We report RV as mean of binary judgments:

$$\text{RV} = \mathbb{E}\Big[J_{\text{LLM}}\big(\hat{\rho}, \mathcal{R}^\star(\hat{P}, \mathbf{a})\big)\Big].$$

Composite CapNav Score. We report a composite score that aggregates the above aspects:

$$\text{CapNav} = \lambda_c F_1 + \lambda_p \text{PV} + \lambda_t \overline{\text{RTA}} + \lambda_r \overline{\text{RV}}, \sum \lambda. = 1,$$

where $\overline{\text{RTA}}$ averages over positive predictions, and $\overline{\text{RV}}$ averages over negative predictions. We report per-embodiment scores and a macro-average across embodiments. By default we set all λ as 0.25.

4. Experiments

Our experiments address three research questions (RQs):

1. **RQ1:** How do state-of-the-art (SOTA) vision–language models (VLMs) perform on the CapNav task?
2. **RQ2:** How does performance vary across models, input frame rates, and agent types?
3. **RQ3:** What failure modes reveal the current limitations of these models?

To address these questions, we evaluate 13 modern VLMs on CapNav. We analyze performance across model families/types, input frame rates, agent embodiments, and challenge types. We also examine common failure cases to diagnose sources of error.

4.1. Models

We evaluate 13 VLMs, including popular proprietary and open-sourced models like the Gemini 2.5 family, GPT, Doubao-Seed, and Qwen3-VL. Some models (*e.g.,* GLM-4.1V, GPT-5-pro) have built-in "thinking" modes by default, while others (e.g., Intern-VL3.5, MiMo-VL) offer "thinking" as an option. For models with an optional setting, we report results for both *thinking* and *non-thinking* configurations. Because CapNav requires spatial reasoning, we additionally include two models, Spatial-MLLM [52] and Vide-R1 [12], that explicitly target spatial reasoning.

At inference time, each model receives a CapNav instance encoded as a triple $\langle \mathcal{S}, \tau, \mathbf{a} \rangle$ (see Section 3.1). Depending on model I/O constraints, $\mathcal{S}$ is provided either as a

CVPR #2124

Model	Feas-F1	PV	RTA	RV	CapNav	HUMAN	HUMANOID	QUAD	SWEEP	WHEEL
Non-thinking mode										
Intern-VL3.5-8B [62]	73.65	26.53	39.54	27.17	41.72	53.89	35.61	51.92	51.36	49.43
Intern-VL3.5-14B [62]	75.68	35.82	50.21	25.18	46.72	50.43	42.98	47.69	44.66	47.38
Keye-VL-1.5-8B [56]	77.63	30.84	43.96	36.70	47.28	58.64	**53.44**	54.37	**62.17**	60.11
MiMo-VL-7B-RL-2508 [46]	57.31	43.61	60.08	29.35	47.59	46.61	38.96	45.38	41.46	50.85
Qwen3-VL-8B-Instruct [47]	78.92	43.39	53.56	**47.89**	55.94	54.20	38.18	50.91	57.50	53.21
GPT-4.1 [35]	75.90	49.00	65.51	32.86	55.82	**73.49**	40.35	49.74	57.70	56.24
Doubao-Seed-1.6 [41]	**80.26**	**60.00**	**71.92**	35.47	**61.91**	65.18	47.23	**59.49**	60.65	**60.82**
Thinking mode										
Spatial-MLLM-4B [52]	75.27	5.04	10.16	-	30.15	35.87	14.86	29.48	25.78	22.69
Video-R1-7B [12]	74.57	25.50	44.66	4.82	37.39	43.62	26.35	41.66	32.17	31.82
Keye-VL-1.5-8B [56]	73.07	41.61	48.54	38.67	50.47	51.05	30.99	47.59	46.52	45.98
GLM-4.1V-9B [48]	73.81	40.12	55.43	33.12	50.62	50.90	33.19	46.01	45.47	42.46
Intern-VL3.5-14B [62]	78.48	42.96	57.90	24.97	51.08	50.21	39.93	51.13	46.27	47.67
Intern-VL3.5-8B [62]	79.55	43.74	55.10	36.50	53.72	51.23	35.34	51.61	53.74	50.05
MiMo-VL-7B-RL-2508 [46]	65.10	59.62	65.00	35.33	56.26	46.61	38.96	45.38	41.46	50.85
Doubao-Seed-1.6 [41]	76.16	61.94	71.93	**38.44**	62.12	60.31	46.20	58.05	61.02	63.35
Gemini-2.5-flash [10]	84.16	65.12	68.96	38.04	64.07	79.95	40.21	59.15	63.07	59.46
GPT-5-pro [36]	**86.87**	67.90	75.89	34.81	66.37	82.85	46.95	**61.74**	70.58	64.78
Gemini-2.5-pro [10]	84.30	**73.00**	**79.15**	32.29	**67.18**	**85.96**	**54.47**	61.21	**70.87**	**65.51**

Table 1. CapNav performance across 13 VLMs. Each number indicate the best performance under all tested frame settings. Left block reports each task metrics (Feas-F1, PV, RTA, RV, CapNav). Right block reports per-agent composite scores for the five agent types: adult with no motor disabilities, humanoid robot, quadrupedal, sweeping robots, and wheelchair users. Bold number indicate best per block. Blue indicate open-sourced model; Green indicate proprietary; Orange indicate spatial reasoning models.

video clip or as a list of sampled frames. A practical bottleneck for current VLMs is the maximum number of visual tokens/frames accepted per query. Some APIs (e.g., Gemini, Qwen) allow relatively large frame budgets, whereas others (e.g., GPT and smaller open-sourced models) are limited to ∼16–64 frames.

To study this input constraint, we define four input frame rates and test each model on as many frame rates as its interface permits: **16**, **32**, and **64** frames per clip, plus a **1 FPS** coarsely sampled version of the full video.

4.2. Performance

Table 1 summarizes the full benchmark results across all evaluated models under both non-thinking and thinking settings. The strongest performance is with proprietary models, and with thinking mode. Gemini-2.5-pro, GPT-5-pro, and Gemini-2.5-flash consistently achieve the highest feasibility F1, path-validity, and route-traversability scores. Seed-1.6 also performs competitively, whereas open-source systems exhibit weaker performance. Spatial reasoning models, including Spatial-MLLM and Video-R1-7B, perform substantially lower on every metric aspect. Overall, the table shows a clear separation between top-tier closed-source models and the rest of the field. It also shows that although spatial reasoning models claim higher performance in spatial tasks, their architecture and training scheme cannot yet generalize on the CapNav benchmark.

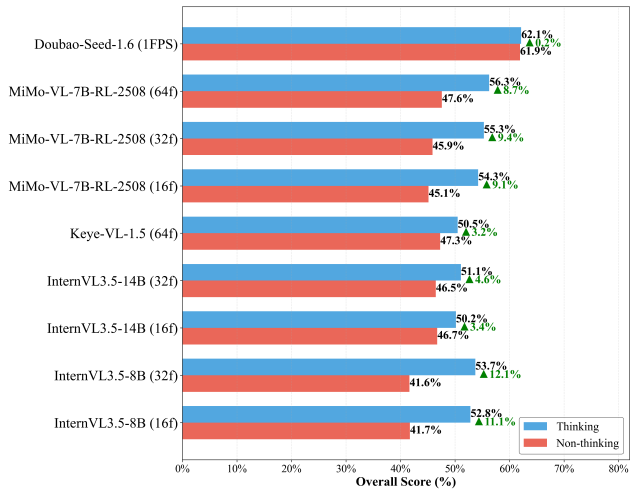

Figure 5. Performance differences between thinking and non-thinking modes for the same models.

Thinking mode. A clear performance boost can be observed in thinking mode. Figure 5 shows models tested under both thinking and non-thinking modes, where a clear increase in overall performance can be seen. Considering that the best performances are also observed among thinking models, it is clear that the CapNav task significantly benefits from thinking capability.

CVPR
#2124

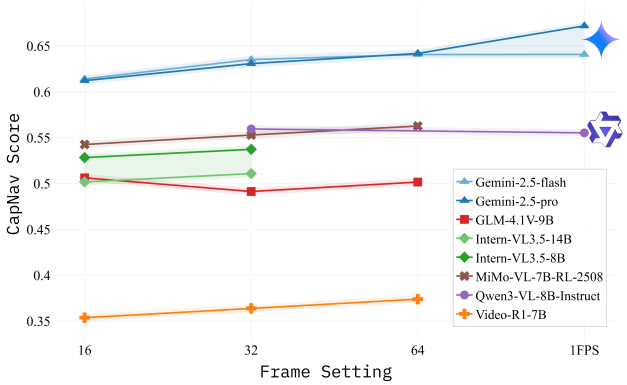

Figure 6. Performance differences across frame settings for the same models

Input frame rate. We next examine the performance changes under different visual input budgets, using 16, 32, and 64 sampled frames per query, as well as a 1FPS version of the full video (Figure 6). Models exhibit non-decreasing trends as the input frames increase. Gemini-2.5-pro and Gemini-2.5-flash show the largest gains. Open-sourced models also benefit from additional frames, though the magnitude of improvement is smaller. These results demonstrate clear budget-dependent differences among model families, indicating that higher frame counts is generally beneficial for the CapNav task but yields uneven gains across different VLMs.

Agent types. Performance also varies substantially across the five agent types. Among all evaluated systems, Gemini-2.5-pro achieves the highest scores on four of the five embodiment types, while GPT-5-pro attains the best results on the QUADRUPEDAL setting. Among the five agent types, HUMAN—adults with no motor disabilities—achieves highest performance, which also serves as a baseline that reflects VLM's navigation performance in CapNav's task setting **without capability constraints**. We observe that the further the agent's capability deviates from HUMAN, the weaker the performance is. For example, the HUMANOID deviates the most from HUMAN, for its inability in stair climbing and a wide $0.9m$ pathway clearance, thus witnessing the lowest performance. This indicates the deficits in the current VLM's navigation ability under mobility constraints.

Main obstacle types. To better understand the performance gaps for CapNav on current VLMs, We further stratify by major obstacle categories in the traversability ground truth. We observe four major types of obstacles: **stairs**, which block humanoid robots, sweeping robots, and wheelchair users; **door sill/floor height differences**, which stop wheelchair users and sweeping robots; **narrow pathways**, usually caused by narrow doors or furniture clutter, which can restrict humanoid robots, and also wheelchair users in extreme cases; **lacking of turning spaces**, which hinders wheelchair users and, in rare cases, quadrupedal robots. For the navigation tasks labeled as non-traversable due to one or more of these obstacles, we evaluate how often VLMs can both correctly predict infeasibility and provide appropriate reasoning.

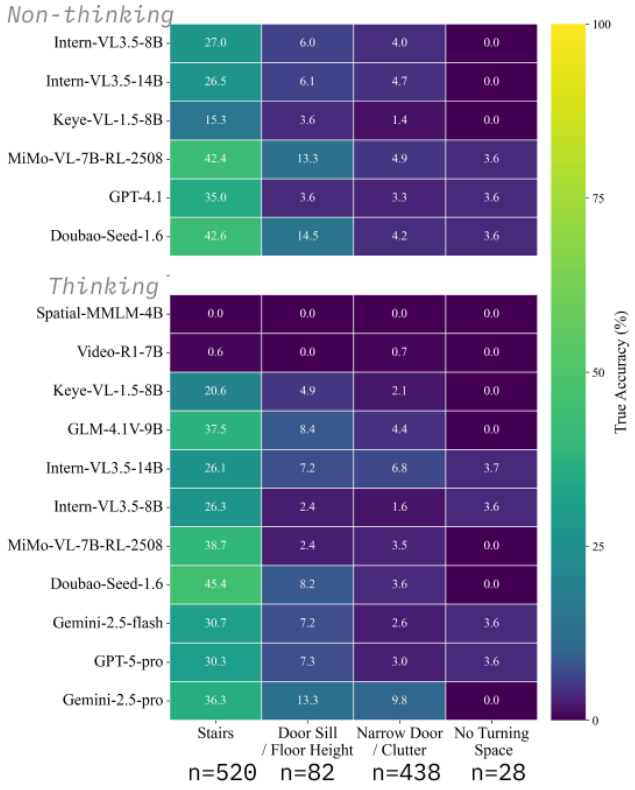

Figure 7. Model accuracy by obstacle type. Each cell shows the percentage of VLM responses that has both correct "*Cannot Go*" predictions and also correct reasoning.

Figure 7 summarizes model performance across these obstacle types. Overall, we observe consistently low accuracy across all models. Among the categories, obstacles with qualitative, visually salient cues (e.g., the presence of stairs or a door sill) yield relatively higher performance, whereas obstacles requiring quantitative spatial estimation (e.g., narrow passages or required turning radii) remain substantially more difficult. This suggests that, although models exhibit understanding of simple capability constraints (e.g., "wheelchair users cannot climb stairs"), reliable traversability judgments that depend on precise spatial measurements inferred from visual tokens remain challenging—even for state-of-the-art VLMs.

CVPR #2124

5. Discussion

We discuss the implications of our findings, outline practical guidelines for applying VLMs to CapNav, and propose directions for future work.

5.1. Summary of Key Findings

Our evaluation reveals substantial performance disparities across model families, agent embodiments, obstacle types, and input configurations. Closed-source models—particularly Gemini-2.5-pro, GPT-5-pro, and Doubao-Seed—form the top tier across nearly all metrics. In contrast, open-source VLMs, including models explicitly designed for spatial reasoning, lag significantly behind.

Performance improves with larger model sizes, thinking-mode inference, and higher input frame rates, indicating that both representational capacity and intermediate reasoning benefit CapNav. However, these gains come with higher computational costs and latency, making them non-trivial to deploy in real-time robotics or resource-constrained systems. Moreover, the magnitude of improvement varies noticeably across model families, highlighting the need for per-model validation rather than assuming uniform benefits.

A central finding is the consistent degradation of performance when the agent's mobility constraints deviate from human norms. Even state-of-the-art models that perform strongly for default settings struggle with the constraints imposed by humanoid robots, sweeping robots, and wheelchair users. This illustrates a critical limitation of applying general-purpose VLMs to specialized embodiments: strong performance in human-like settings does not translate to capability-constrained scenarios, and targeted evaluation is essential before deployment in safety-critical applications.

Our obstacle-type analysis further shows that VLMs remain highly unreliable in tasks that require reasoning on spatial dimension, such as assessing doorway clearance or turning radii. They struggle with fine-grained geometric judgments derived from visual tokens. Consequently, even the strongest models do not yet yield dependable traversability assessments for agents that require passage clearance.

Taken together, these findings call for future research into improving spatial-dimension reasoning, better utilization of long-horizon visual inputs, and architectures that can robustly integrate geometric scene information—capabilities that are not yet well supported by current multimodal LLMs.

5.2. Usability Across Capabilities

The five agent types supported in CapNav are chosen to represent common human and robotic mobility profiles. As such, our results generalize to a wide range of practical embodiments. Practitioners may approximate expected model performance for their own systems by aligning their agent's capabilities with the closest of the five provided profiles.

For embodiments that differ substantially—for instance, robots with unusual locomotion modes or devices with atypical clearance or actuation constraints—CapNav's open-sourced annotation interface and data curation pipeline allow users to define new agent profiles and extend the benchmark. This facilitates capability-aware evaluation for diverse hardware platforms and emerging robotic systems.

5.3. Further Use of the CapNav Dataset

Beyond benchmarking, CapNav provides a valuable resource for improving VLMs' spatial reasoning. Its combination of video tours, structured traversability graphs, and capability-conditioned QA pairs enables several research directions:

- Finetuning or instruction-tuning VLMs for embodied navigation and spatial understanding.
- Augmenting pretraining corpora with explicit geometric and capability-aware supervision.
- Programmatically generating new navigation QA tasks using the graph structure to increase training coverage.
- Developing hybrid reasoning pipelines that combine VLM perception with geometric or classical planning modules.

These avenues can help narrow the gap towards capability-aware navigation, moving toward VLMs that can reason reliably about embodiment-specific mobility constraints in real-world spaces.

6. Conclusion

We present CapNav, a benchmark that systematically evaluates capability-conditioned navigation across diverse embodiments, realistic mobility obstacles, and graph-structured indoor environments. By assessing 13 state-of-the-art VLMs over 45 scenes, 2365 navigation tasks, and more than 5,000 traversability annotations, we show that while modern VLMs exhibit strong navigation performance in human-like settings, they fail to generalize when physical capabilities diverge. Performance degrades for more constrained agents, for obstacle types requiring quantitative spatial reasoning, and under limited visual input budgets—revealing gaps that current VLMs can only partially mitigate. These findings underscore the risks of applying general-purpose VLMs as drop-in navigation planners without capability-aware evaluation, and highlight the need for models that reason robustly about physical dimensions, environmental constraints, and embodiment-specific affordances. By releasing the benchmark, dataset, and annotation interfaces, CapNav establishes a foundation for developing and evaluating next-generation VLMs that support safer, more inclusive, and capability-aware navigation in complex real-world spaces.

References

[1] Peter Anderson and et al. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, 2018. 1, 2, 3, 5

[2] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. Objectnav revisited: On evaluation of embodied agents navigating to objects. *arXiv preprint arXiv:2006.13171*, 2020. 3

[3] Cătălina Cangea, Eugene Belilovsky, Pietro Liò, and Aaron Courville. Videonavqa: Bridging the gap between visual and embodied question answering. *arXiv preprint arXiv:1908.04950*, 2019. 3

[4] Mateo Guaman Castro, Sidharth Rajagopal, Daniel Gorbatov, Matt Schmittle, Rohan Baijal, Octi Zhang, Rosario Scalise, Sidharth Talia, Emma Romig, Celso de Melo, et al. Vamos: A hierarchical vision-language-action model for capability-modulated and steerable navigation. *arXiv preprint arXiv:2510.20818*, 2025. 3

[5] Marvin Chancán and Michael Milford. Citylearn: Diverse real-world environments for sample-efficient navigation policy learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1697–1704. IEEE, 2020. 3

[6] Angel Chang and et al. Matterport3d: Learning from rgb-d data in indoor environments. In *3DV*, 2017. 2, 4

[7] Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12538–12547, 2019. 3

[8] Sirui Chen, Yufei Ye, Zi-ang Cao, Jennifer Lew, Pei Xu, and C Karen Liu. Hand-eye autonomous delivery: Learning humanoid navigation, locomotion and reaching. *arXiv preprint arXiv:2508.03068*, 2025. 3

[9] 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. *Advances in Neural Information Processing Systems*, 37:135062–135093, 2024. 1

[10] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 6

[11] Jean Marc Feghali, Cheng Feng, Arnab Majumdar, and Washington Yotto Ochieng. Comprehensive review: High-performance positioning systems for navigation and wayfinding for visually impaired people. *Sensors*, 24(21): 7020, 2024. 3

[12] Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. *arXiv preprint arXiv:2503.21776*, 2025. 1, 5, 6

[13] Rita Fleiner, Gabriella Simon-Nagy, and Barnabás Szász. Accessible indoor navigation based on linked data in hospitals. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 002425–002430. IEEE, 2016. 3

[14] Dylan Goetting, Himanshu Gaurav Singh, and Antonio Loquercio. End-to-end navigation with vision language models: Transforming spatial reasoning into question-answering. *arXiv preprint arXiv:2411.05755*, 2024. 1

[15] Tianrui Guan and et al. Hallusionbench: An advanced diagnostic suite for entangled language hallucination and visual illusion in lvlms. In *CVPR*, 2024. 2

[16] David Hoeller, Nikita Rudin, Dhionis Sako, and Marco Hutter. Anymal parkour: Learning agile navigation for quadrupedal robots. *Science Robotics*, 9(88):eadi7566, 2024. 3

[17] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. *arXiv preprint arXiv:2210.05714*, 2022. 1

[18] Minyoung Hwang, Jaeyeon Jeong, Minsoo Kim, Yoonseon Oh, and Songhwai Oh. Meta-explore: Exploratory hierarchical vision-and-language navigation using scene object spectrum grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6683–6693, 2023. 1

[19] Vihan Jain, Gabriel Magalhaes, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. Stay on the path: Instruction fidelity in vision-and-language navigation. *arXiv preprint arXiv:1905.12255*, 2019. 3

[20] Md Mohsin Kabir, Jamin Rahman Jim, and Zoltan Istenes. Terrain detection and segmentation for autonomous vehicle navigation: A state-of-the-art systematic review. *Information Fusion*, 113:102644, 2025. 3

[21] Mukul Khanna, Ram Ramrakhya, Gunjan Chhablani, Sriram Yenamandra, Theophile Gervet, Matthew Chang, Zsolt Kira, Devendra Singh Chaplot, Dhruv Batra, and Roozbeh Mottaghi. Goat-bench: A benchmark for multi-modal lifelong navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16373–16383, 2024. 3

[22] Jing Yu Koh, Honglak Lee, Yinfei Yang, Jason Baldridge, and Peter Anderson. Pathdreamer: A world model for indoor navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14738–14748, 2021. 1

[23] Jacob Krantz and et al. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *ECCV*, 2020. 2

[24] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. *arXiv preprint arXiv:2010.07954*, 2020. 3

[25] Joonho Lee, Marko Bjelonic, Alexander Reske, Lorenz Wellhausen, Takahiro Miki, and Marco Hutter. Learning robust autonomous navigation and locomotion for wheeled-legged robots. *Science Robotics*, 9(89):eadi9641, 2024. 3

[26] Chu Li, Rock Yuren Pang, Delphine Labbé, Yochai Eisenberg, Maryam Hosseini, and Jon E Froehlich. Accessibility

for whom? perceptions of mobility barriers across disability groups and implications for designing personalized maps. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–19, 2025. 3

[27] Jialu Li, Aishwarya Padmakumar, Gaurav Sukhatme, and Mohit Bansal. Vln-video: Utilizing driving videos for outdoor vision-and-language navigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 18517–18526, 2024. 3

[28] Xiao Li, Bharat Gandhi, Ming Zhan, Mohit Nehra, Zhicheng Zhang, Yuchen Sun, Meijia Song, Naisheng Zhang, and Xi Wang. Fine-tuning vision-language models for visual navigation assistance. *arXiv preprint arXiv:2509.07488*, 2025. 1

[29] Yiyou Li and et al. Evaluating object hallucination in large vision-language models. *arXiv:2305.10355*, 2023. 2

[30] Hongyi Liu and Quan Yuan. Safe and robust motion planning for autonomous navigation of quadruped robots in cluttered environments. *IEEE Access*, 12:69728–69737, 2024. 3

[31] Xinhao Liu, Jintong Li, Yicheng Jiang, Niranjan Sujay, Zhicheng Yang, Juexiao Zhang, John Abanes, Jing Zhang, and Chen Feng. Citywalker: Learning embodied urban navigation from web-scale videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6875–6885, 2025. 3

[32] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16488–16498, 2024. 3

[33] Piotr Mirowski, Andras Banki-Horvath, Keith Anderson, Denis Teplyashin, Karl Moritz Hermann, Mateusz Malinowski, Matthew Koichi Grimes, Karen Simonyan, Koray Kavukcuoglu, Andrew Zisserman, et al. The streetlearn environment and dataset. *arXiv preprint arXiv:1903.01292*, 2019. 3

[34] Gabriel Iluebe Okolo, Turke Althobaiti, and Naeem Ramzan. Assistive systems for visually impaired persons: challenges and opportunities for navigation assistance. *Sensors*, 24(11): 3572, 2024. 3

[35] OpenAI. Introducing gpt-4.1 in the api. https://openai.com/index/gpt-4-1/, 2025. Accessed: 2025-11-12. 6

[36] OpenAI. Introducing gpt-5. https://openai.com/index/introducing-gpt-5/, 2025. Accessed: 2025-11-12. 6

[37] Yuankai Qi and et al. Reverie: Remote embodied visual referring expression in real indoor environments. In *CVPR*, 2020. 3

[38] Santhosh K. Ramakrishnan and et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv:2109.08238*, 2021. 2, 4

[39] Sushil Kumar Sahoo and Bibhuti Bhusan Choudhury. Autonomous navigation and obstacle avoidance in smart robotic wheelchairs. *Journal of Decision Analytics and Intelligent Computing*, 4(1):47–66, 2024. 3

[40] Manolis Savva and et al. Habitat: A platform for embodied ai research. In *ICCV*, 2019. 2, 5

[41] ByteDance Seed. Seed 1.6. https://seed.bytedance.com/en/seed1_6/, 2025. Accessed: 2025-11-12. 6

[42] Dhruv Shah and et al. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In *CoRL (PMLR)*, 2023. 1

[43] Xiangxi Shi, Zhonghua Wu, and Stefan Lee. Aware visual grounding in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14056–14065, 2024. 1

[44] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749, 2020. 3

[45] Xinshuai Song, Weixing Chen, Yang Liu, Weikai Chen, Guanbin Li, and Liang Lin. Towards long-horizon vision-language navigation: Platform, benchmark and method. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12078–12088, 2025. 1

[46] Core Team, Zihao Yue, Zhenru Lin, Yifan Song, Weikun Wang, Shuhuai Ren, Shuhao Gu, Shicheng Li, Peidian Li, Liang Zhao, Lei Li, Kainan Bao, Hao Tian, Hailin Zhang, Gang Wang, Dawei Zhu, Cici, Chenhong He, Bowen Ye, Bowen Shen, Zihan Zhang, Zihan Jiang, Zhixian Zheng, Zhichao Song, Zhenbo Luo, Yue Yu, Yudong Wang, Yuanyuan Tian, Yu Tu, Yihan Yan, Yi Huang, Xu Wang, Xinzhe Xu, Xingchen Song, Xing Zhang, Xing Yong, Xin Zhang, Xiangwei Deng, Wenyu Yang, Wenhan Ma, Weiwei Lv, Weiji Zhuang, Wei Liu, Sirui Deng, Shuo Liu, Shimao Chen, Shihua Yu, Shaohui Liu, Shande Wang, Rui Ma, Qiantong Wang, Peng Wang, Nuo Chen, Menghang Zhu, Kangyang Zhou, Kang Zhou, Kai Fang, Jun Shi, Jinhao Dong, Jiebao Xiao, Jiaming Xu, Huaqiu Liu, Hongshen Xu, Heng Qu, Haochen Zhao, Hanglong Lv, Guoan Wang, Duo Zhang, Dong Zhang, Di Zhang, Chong Ma, Chang Liu, Can Cai, and Bingquan Xia. Mimo-vl technical report, 2025. 6

[47] Qwen Team. Qwen3 technical report, 2025. 6

[48] V Team, Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, Shuaiqi Duan, Weihan Wang, Yan Wang, Yean Cheng, Zehai He, Zhe Su, Zhen Yang, Ziyang Pan, Aohan Zeng, Baoxu Wang, Bin Chen, Boyan Shi, Changyu Pang, Chenhui Zhang, Da Yin, Fan Yang, Guoqing Chen, Jiazheng Xu, Jiale Zhu, Jiali Chen, Jing Chen, Jinhao Chen, Jinghao Lin, Jinjiang Wang, Junjie Chen, Leqi Lei, Letian Gong, Leyi Pan, Mingdao Liu, Mingde Xu, Mingzhi Zhang, Qinkai Zheng, Sheng Yang, Shi Zhong, Shiyu Huang, Shuyuan Zhao, Siyan Xue, Shangqin Tu, Shengbiao Meng, Tianshu Zhang, Tianwei Luo, Tianxiang Hao, Tianyu Tong, Wenkai Li, Wei Jia, Xiao Liu, Xiaohan Zhang, Xin Lyu, Xinyue Fan, Xuancheng Huang, Yanling Wang, Yadong Xue, Yanfeng Wang, Yanzi Wang, Yifan An, Yifan Du, Yiming Shi, Yiheng Huang, Yilin Niu,

Yuan Wang, Yuanchang Yue, Yuchen Li, Yutao Zhang, Yuting Wang, Yu Wang, Yuxuan Zhang, Zhao Xue, Zhenyu Hou, Zhengxiao Du, Zihan Wang, Peng Zhang, Debing Liu, Bin Xu, Juanzi Li, Minlie Huang, Yuxiao Dong, and Jie Tang. Glm-4.5v and glm-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning, 2025. 6

[49] Arun Balajee Vasudevan, Dengxin Dai, and Luc Van Gool. Talk2nav: Long-range vision-and-language navigation in cities. *CoRR*, 2019. 3

[50] Martin Weiss, Simon Chamorro, Roger Girgis, Margaux Luck, Samira E. Kahou, Joseph P. Cohen, Derek Nowrouzezahrai, Doina Precup, Florian Golemo, and Chris Pal. Navigation agents for the visually impaired: A sidewalk simulator and experiments, 2019. 3

[51] Tim Windecker, Manthan Patel, Moritz Reuss, Richard Schwarzkopf, Cesar Cadena, Rudolf Lioutikov, Marco Hutter, and Jonas Frey. Navitrace: Evaluating embodied navigation of vision-language models. *arXiv preprint arXiv:2510.26909*, 2025. 3

[52] Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. Spatial-mllm: Boosting mllm capabilities in visual-based spatial intelligence. *arXiv preprint arXiv:2505.23747*, 2025. 1, 5, 6

[53] Wayne Wu, Honglin He, Jack He, Yiran Wang, Chenda Duan, Zhizheng Liu, Quanyi Li, and Bolei Zhou. Metaurban: An embodied ai simulation platform for urban micromobility. *arXiv preprint arXiv:2407.08725*, 2024. 3

[54] Wayne Wu, Honglin He, Chaoyuan Zhang, Jack He, Seth Z Zhao, Ran Gong, Quanyi Li, and Bolei Zhou. Towards autonomous micromobility through scalable urban simulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 27553–27563, 2025.

[55] Ziyang Xie, Zhizheng Liu, Zhenghao Peng, Wayne Wu, and Bolei Zhou. Vid2sim: Realistic and interactive simulation from video for urban navigation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1581–1591, 2025. 3

[56] Biao Yang, Bin Wen, Boyang Ding, Changyi Liu, Chenglong Chu, Chengru Song, Chongling Rao, Chuan Yi, Da Li, Dunju Zang, et al. Kwai keye-vl 1.5 technical report. *arXiv preprint arXiv:2509.01563*, 2025. 6

[57] 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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10632–10643, 2025. 1

[58] Zecheng Yin, Chonghao Cheng, et al. Navigation with vlm framework: Go to any language. *arXiv preprint arXiv:2410.02787*, 2024. 1

[59] Xiaochen Zhang, Ziyang Song, Qianbo Huang, Ziyi Pan, Wujing Li, Ruining Gong, and Bi Zhao. Shared ehmi: Bridging human–machine understanding in autonomous wheelchair navigation. *Applied Sciences*, 14(1):463, 2024. 3

[60] Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. *arXiv:2305.16986*, 2023. 1

[61] Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10012–10022, 2020. 3

[62] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025. 6