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

Xia Su*
University of Washington
Seattle, WA, USA
xiasu@cs.washington.edu

Ruiqi Chen*
University of Washington
Seattle, WA, USA
ruiqich@uw.edu

Benlin Liu
University of Washington
Seattle, WA, USA
liubl@cs.washington.edu

Jingwei Ma
University of Washington
Seattle, WA, USA
jingweim@cs.washington.edu

Zonglin Di
University of California, Santa Cruz
Santa Cruz, CA, USA
zdi@ucsc.edu

Ranjay Krishna
University of Washington
Seattle, WA, USA
ranjay@cs.washington.edu

Jon Froehlich
University of Washington
Seattle, WA, USA
jonf@cs.washington.edu

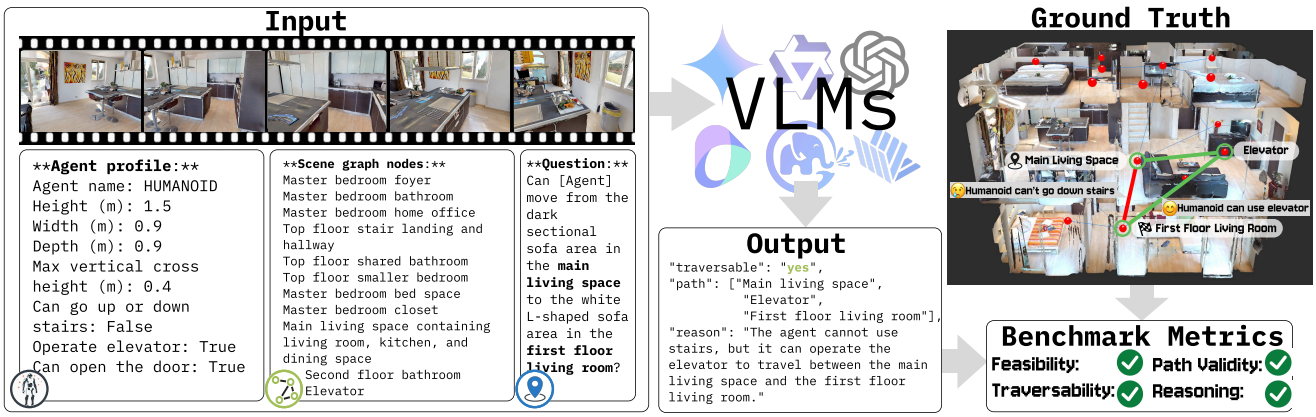

Figure 1. We introduce Capability-Conditioned Navigation (***CapNav***), a benchmark designed to evaluate how well VLMs can navigate complex indoor spaces given an agent's specific physical and operational capabilities. CapNav inputs (1) a tour video of an indoor space, (2) nodes of its navigation graph, (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 conclude by discussing the implications for capability-aware navigation and the opportunities for advancing embodied spatial reasoning in future VLMs. The benchmark is available at https://github.com/makeabilitylab/CapNav

1. Introduction

As vision–language models (VLMs) advance in visual grounding [47] and spatial reasoning [11], they are now commonly used as drop-in "navigation assistants" or planning modules for robots and assistive systems [16, 46, 66]. Guiding people to reach destinations [16, 30] and controlling robots in movement decisions [20], they are widely applied in open-ended scenarios where solutions can be pluralistic, and their plausibility depends on the agents' movement capabilities.

Consider the multi-story building shown in Figure 2. To go from the basement foyer to the top-floor rest area, a wheelchair user should use the elevator, but a quadruped robot can only go up 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 awareness of capability, VLM may recommend routes that are infeasible or even unsafe for a given agent [17, 31].

Such mobility constraints and the pluralistic nature of navigation remain underexplored in prior vision–language navigation research. Existing frameworks and benchmarks evaluate embodiment-agnostic goal-reaching in simulated or graph-based environments [2, 25, 49, 54], or reduce navigation to simplified VQA-style spatial reasoning tasks [14, 57, 61, 63]. Moreover, evaluation protocols typically measure trajectory fidelity against a single annotated ground-truth path [1, 21, 22]. Consequently, these paradigms overlook the capability-conditioned validity and solution multiplicity that characterize real-world navigation — limitations that become increasingly critical as VLMs are deployed in embodied control and assistive scenarios.

We introduce *CapNav*, a benchmark for VLM's navigation validity across varying mobility constraints. We consider five representative embodiments: adults with no motor disabilities, wheelchair users, sweeping robots, humanoid robots, and quadrupedal robots. We use real-world 3D scans [8, 41] and annotate navigation graphs [2, 25, 44] to allow plural solutions and per-edge traversability analysis. CapNav inputs the agent description, realistic indoor videos, key spatial reference nodes, and a navigation task as"from A to B". We then examine VLM outputs for the prediction of feasibility, the navigation path, and reasoning validity. In total, CapNav contains 45 scenes, 2365 navigation tasks, and 5075 traversability annotations as ground truth.

Using the CapNav benchmark, we systematically study 13 opensource and proprietary VLMs under varying inference settings. Through quantitative and qualitative analysis, we diagnose three major observations: (1) **mobility degradation**: navigation performance drops significantly when considering mobility capabilities; (2) **visual bottleneck**: increasing visual input does not always lead to a performance boost in CapNav; (3) **spatial dimension blindness**: all models fail to detect non-traversability arising from insufficient spatial clearance (e.g., narrow passages or limited turning radius).

These results highlight that substantial progress is still needed before VLM-based navigation can reliably support embodiment-specific deployment, particularly in safety-critical scenarios. Moreover, current VLM architectures and training paradigms remain limited in their ability to integrate spatial information across multiple visual frames and to perform precise metric reasoning in complex environments. Advancing embodied navigation, therefore, requires improvements in large-scale visual integration, geometry-grounded reasoning, and training objectives that explicitly encode mobility constraints and spatial dimension awareness.

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.

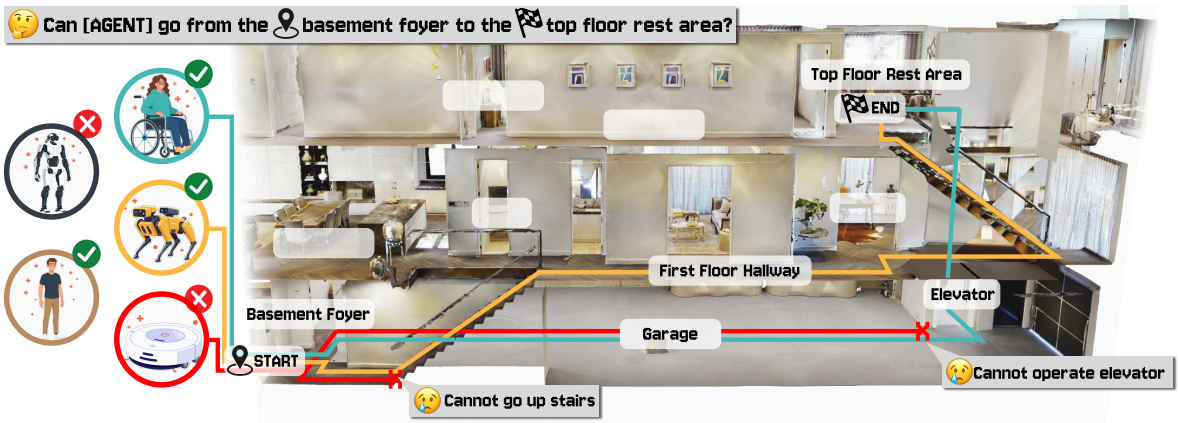

Figure 2. The CapNav benchmark evaluates whether VLMs can correctly ground differences in agent mobility capabilities when generating navigation plans. This example demonstrates a navigation task that has different feasibility and path for different agents.

2. Related Work

2.1. Vision Language Navigation and Benchmarks

Vision-Language Navigation (VLN) studies how an agent follows natural-language instructions using grounded visual observations to traverse an environment and reach a goal [2, 67]. Since the introduction of Room-to-Room (R2R) [2], which casts indoor navigation as step-by-step action selection on a sparse connectivity graph, a wide range of datasets and benchmarks have expanded VLN along multiple axes. Indoor instruction-following has been scaled in path length, diversity, and grounding density (e.g., R4R [22] and RxR [26]); broader urban/street settings have also been explored (e.g., TOUCHDOWN [9], StreetLearn [35], CityLearn [7], CityWalker [33], Talk2Nav [53]). Other benchmarks couple navigation with object grounding or task execution (e.g., REVERIE [39], ALFRED [48], ObjectNav [3], EmbodiedBench [64]), while newer efforts probe long-horizon and open-ended embodied behaviors (e.g., GOAT-bench [24] and NavBench [40]).

Beyond task variants, VLN benchmarks differ in their observation and interaction assumptions. The canonical setting is *interactive*: an agent explores incrementally, receiving egocentric observations and accumulating spatial knowledge step-by-step. In contrast, a *passive* formulation provides global context, such as panoramic scans or pre-recorded videos, aiming to evaluate high-level spatial reasoning while reducing confounds from exploration and low-level control. For example, VideoNavQA [5] replaces interactive navigation with oracle trajectory videos to isolate embodied visual reasoning; NRNS [18] learns image-goal navigation from passive roaming videos without online RL or a simulator; and PONI [42] trains an object-search module interaction-free from offline semantic maps. OpenEQA [34] further supports an episodic-memory (video) setting that evaluates spatial memory and reasoning over recorded tours and scans. Under such passive formulations, the task increasingly resembles route planning and feasibility assessment; following prior work [5, 34], we still refer to this setting as navigation.

Our goal—benchmarking VLM navigation validity under explicit mobility constraints—benefits from global observation, since it allows models to reason about traversability and alternative routes across the entire scene while isolating embodiment constraints from exploration and execution noise. We further adopt a graph-based spatial abstraction [2, 26] to support pluralistic solutions and enable fine-grained evaluation of path feasibility and validity. This complements existing VLN benchmarks by enabling structured, capability-conditioned comparisons across real-world environments and embodiments.

2.2. Navigation with mobility constraints

Past navigation research has modeled a wide range of mobility constraints, including accessibility for wheelchair users [15, 43, 65], assistance for people who are blind or low vision [13, 36, 55], and robotics embodiments such as quadrupeds [19, 27, 32], humanoids [10], and vehicles operating over varied terrain [23, 29]. Some works also consider settings where multiple mobility profiles co-exist, either through simulator diversity [58–60] or by modeling user preferences [28]. Collectively, these studies highlight that embodiment constraints materially change feasible routes and navigation strategies.

However, most evaluations remain siloed—focusing on a single embodiment, environment family, or constraint type—making cross-embodiment comparison difficult on matched tasks. Recent efforts begin to address this gap: VLN-PE (introduced with a holistic study of physical and visual disparities) enables systematic cross-embodiment evaluation in a physically grounded simulator setting [54]; NaviTrace [56] evaluates embodiment-conditioned naviga-

tion via 2D trace prediction in real-world images; and VAMOS [6] introduces an affordance-grounded hierarchical planner to modulate plans by embodiment capabilities. Nonetheless, existing benchmarks either emphasize simulator-centric embodiment effects or cast the problem as trace/plan prediction without explicitly validating pluralistic, capability-conditioned route feasibility in complex indoor spaces with realistic navigation obstacles (e.g., multi-floor transitions, bottlenecks, and accessibility-specific affordances).

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 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 condition where all indoor routes can be achieved; **Wheelchair users** cannot use stairs and require clearance for passing and turning; **Humanoid robot** cannot go up/down stairs and require clearance for passing; **Sweeping robots** require a flat floor surface; **Quadrupedal robots** can traverse most indoor spaces but cannot operate doors/elevators.

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. These attributes directly affect traversability on edges $\mathcal{E}$ and on overall tasks τ. See Figure 4 for examples.

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 through each possible simple path that connects v_{src} and v_{tgt} in $\mathcal{G}$, and assign $g_e^{(\mathbf{a})} \in \{0, 1\}$, 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 annotated as text descriptions (*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 the existence of 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:

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

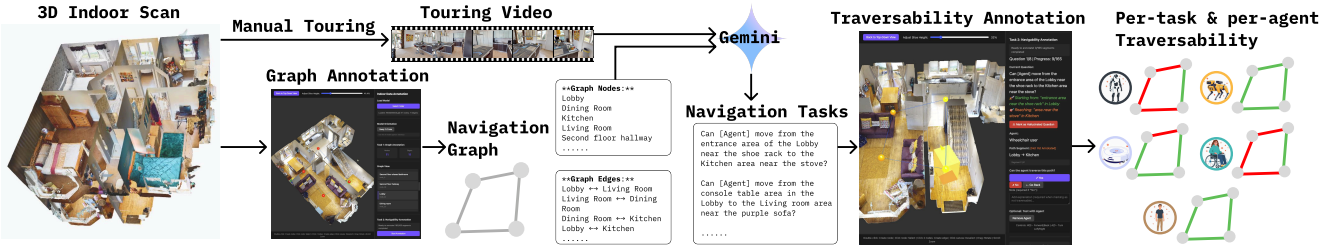

Figure 3. Overview of CapNav's data construction: Starting from a 3D indoor scan, we manually record a touring video and a navigation graph. We then use Gemini to generate natural language navigation tasks. Finally, per-task and per-agent traversability are annotated by manually controlling agents in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

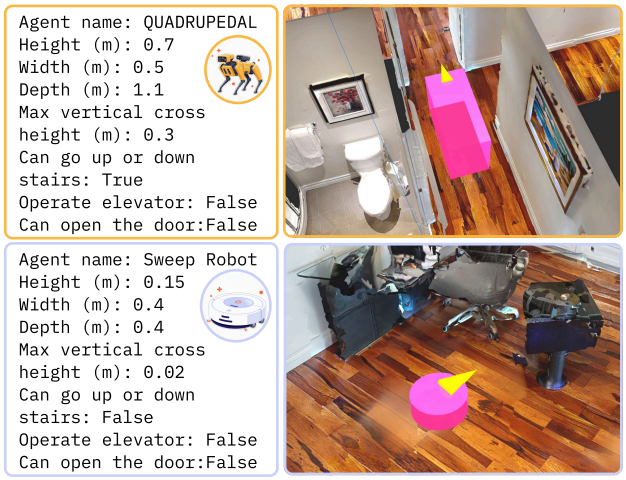

Figure 4. Examples of CapNav's agent profiles. Left: each profile specifies the physical dimensions and functional capabilities. Right: the agents' physical dimensions are rendered and maneuvered in the 3D scenes to help confirm traversability.

3.3. Dataset

We collect 3D indoor scenes from HM3D [41] and Matterport3D [8], and record touring videos in the Habitat simulator [44] by manually navigating each space. Videos are rendered at 2FPS from human-eye height ($1.5m$) with a 75 field of view to mimic casual handheld walkthroughs. For each scene, we construct a navigation graph via a custom annotation interface: nodes $v \in \mathcal{V}$ are annotated with semantic labels $c(v)$ and approximate 3D positions, and edges $e = (u, v) \in \mathcal{E}$ denote bidirectional, directly walkable connections. We provide each scene's video and node list to Gemini 2.5 Pro to generate navigation tasks, manually verify task validity, and annotate edge-level traversability for five embodiments, including reasons for non-traversable cases. After filtering out scenes with major holes, disconnected subspaces, repetitive semantics, or trivial layouts, we retain 45 indoor scenes (avg. 160.38 s per video, 13.8 nodes, 14.5 edges), yielding 2,365 navigation tasks and 5,075 traversability labels (3,945 positive, 1,130 negative). Feasibility varies substantially across agent types in both task-level and edge-level ratios (see Table 1).

Table 1. Ratio of feasible task and traversable edges per agent type

Agent Type	Feasible Task Ratio	Edge Traversable Ratio
Human	1	1
Wheelchair	0.48	0.71
Humanoid	0.22	0.43
Quadrupedal	0.97	0.96
Sweeper	0.57	0.79

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. $\mathrm{RTA} = 1$ indicates a fully valid route, while lower values quantify partial failures. This metric serves a role analogous to path-quality measures such as SPL [2], 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 the annotation and 0 otherwise. We also verified the reliability of this function by manually validating 300 randomly sampled VLM reasonings. The results show an 89% alignment between human and LLM verdicts, indicating the precision and effectiveness of the LLM-as-judge method. We report RV as the mean of binary judgments:

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

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

$$\mathrm{CapNav} \;=\; \lambda_c\, F_1 + \lambda_p\, \mathrm{PV} + \lambda_t\, \overline{\mathrm{RTA}} + \lambda_r\, \overline{\mathrm{RV}}, \sum \lambda_\cdot = 1,$$

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

Reference Upper and Lower Bounds. To contextualize model performance, we establish both lower and upper reference bounds under the proposed evaluation metrics. As a lower bound, we simulate a random-walk policy on the navigation graph that ignores scene structure and agent capability constraints. This naive baseline achieves a CapNav score of 0.29. As an approximate upper bound, we evaluate human performance on the same task. We recruited four human participants, each completing 20 randomly sampled navigation tasks using the same input and output format as the models. Human CapNav scores range from 0.49 to 0.75 (*AVG* = 0.61).

4. Experiments

Our experiments address three research questions:

1. **RQ1:** How do state-of-the-art VLMs perform on CapNav?
2. **RQ2:** How does performance vary across models, agent types, and inference settings?
3. **RQ3:** What failure modes expose the current limitations of these models?

To answer these questions, we evaluate 13 modern VLMs on CapNav. We report quantitative results stratified by model family, agent type, inference setting (frame budget and thinking mode), and challenge type. We further analyze common failure cases and consolidate them into a qualitative error taxonomy.

4.1. Settings

We evaluated 13 VLMs in November 2025, spanning popular proprietary and open-source models with strong multimodal capabilities, including the Gemini 2.5 family, GPT 4.1 and 5-Pro, Doubao-Seed, and Qwen3-VL. For models that expose a thinking option, we report results under both *thinking* and *non-thinking* configurations. We additionally include Spatial-MLLM [57] and Video-R1 [14], which explicitly target spatial reasoning—a core requirement of CapNav. Proprietary models are accessed via APIs, while open-source models are deployed on either a single NVIDIA A100 GPU (80GB) or a single A40 GPU (48GB).

At inference time, each model receives CapNav tasks as a triple $\langle \mathcal{S}, \tau, \mathbf{a} \rangle$ (Section 3.1). Depending on model I/O constraints, $\mathcal{S}$ is provided either as a video clip or as a set of uniformly sampled frames. A practical limitation of current VLM interfaces is the maximum number of visual tokens/frames accepted per request. To study this constraint, we evaluate up to four input settings for each model (subject to interface limits): **16**, **32**, and **64** total frames per clip, and a **1 FPS** setting that samples the full video (avg. 163 frames/video).

4.2. Performance

Using the metrics in Section 3.4, we compare model outputs against the ground-truth navigation graph to compute overall and per-agent performance. Table 2 summarizes performance metrics across all models under both *thinking* and *non-thinking* settings. We show the best metrics when multiple input frame settings are tested.

All evaluated models outperform the random-walk baseline (CapNav = 29.35%). The strongest performance is achieved by proprietary models (notably the Gemini-2.5 family and GPT-5-pro), which exceed the human average (CapNav = 60.59%) but remain below the best human performance (CapNav = 74.77%). Overall, results show a clear separation between top-tier closed-source systems and the remainder of the field.

Models explicitly designed for spatial reasoning (Spatial-MLLM and Video-R1-7B) perform substantially worse across all metrics, despite claims of strong spatial performance. This indicates that their spatial reasoning training schemes and architectural modifications are not yet

Model	Feas-F1	PV	RTA	RV	CapNav	HUMAN	HUMANOID	QUAD	SWEEP	WHEEL
Non-thinking mode										
Intern-VL3.5-8B [68]	73.65	26.53	39.54	27.17	41.72	53.89	35.61	51.92	51.36	49.43
Intern-VL3.5-14B [68]	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 [62]	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 [50]	57.31	43.61	60.08	29.35	47.59	46.61	38.96	45.38	41.46	50.85
Qwen3-VL-8B-Instruct [51]	78.92	43.39	53.56	**47.89**	55.94	54.20	38.18	50.91	57.50	53.21
GPT-4.1 [37]	75.90	49.00	65.51	32.86	55.82	**73.49**	40.35	49.74	57.70	56.24
Doubao-Seed-1.6 [45]	**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 [57]	75.27	5.04	10.16	-	30.15	35.87	14.86	29.48	25.78	22.69
Video-R1-7B [14]	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 [62]	73.07	41.61	48.54	38.67	50.47	51.05	30.99	47.59	46.52	45.98
GLM-4.1V-9B [52]	73.81	40.12	55.43	33.12	50.62	50.90	33.19	46.01	45.47	42.46
Intern-VL3.5-14B [68]	78.48	42.96	57.90	24.97	51.08	50.21	39.93	51.13	46.27	47.67
Intern-VL3.5-8B [68]	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 [50]	65.10	59.62	65.00	35.33	56.26	46.61	38.96	45.38	41.46	50.85
Doubao-Seed-1.6 [45]	76.16	61.94	71.93	**38.44**	62.12	60.31	46.20	58.05	61.02	63.35
Gemini-2.5-flash [12]	84.16	65.12	68.96	38.04	64.07	79.95	40.21	59.15	63.07	59.46
GPT-5-pro [38]	**86.87**	67.90	75.89	34.81	66.37	82.85	46.95	**61.74**	70.58	64.78
Gemini-2.5-pro [12]	84.30	**73.00**	**79.15**	32.29	**67.18**	**85.96**	**54.47**	61.21	**70.87**	**65.51**

Table 2. 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.

sufficient to address the challenges posed by CapNav.

We also test the sensitivity of rankings to the CapNav score composition. In addition to equal weighting, we evaluate four alternative schemes that assign weight 0.5 to one component and $1/6$ to each of the remaining components. Kendall's τ between rankings across these schemes averages 0.909, suggesting that conclusions are largely robust to moderate weight changes.

4.3. Failure-mode taxonomy

To characterize systematic errors, we use Gemini-3 to mine recurring error themes from 1,000 randomly sampled inference outputs, and then manually refine these themes into a four-category taxonomy. We then visualize the VLM predicted results of 1500 QA pairs sampled under 10 model settings in 3D scenes to measure the prevalence and distribution of each error type.

- **Path hallucination:** (N = 659 / 1500) the predicted path links non-adjacent nodes, or violates the specified start/end, indicating an incorrect reasoning of the scene connectivity.
- **Obstacle hallucination:** (N = 418) the model reports non-existent blocking obstacles (e.g., "closed doors" or severe clutter), suggesting unreliable spatial interpretation of visual evidence.
- **Dimension neglect:** (N = 191) the model overlooks geometric constraints that block the agent (e.g., narrow doors, tight clearances, cluttered passages), reflecting weak grounding between spatial extent and traversability.
- **Ability hallucination:** (N = 10) the output reasoning contradicts the provided agent profile. This failure is only observed in smaller open-sourced models (e.g., MiMo-VL 7B).

4.4. Physical Obstacle Types

We further stratify by major obstacle categories in the traversability ground truth. We observe four major types of ground-truth obstacles: **stairs** (N = 520 / 2365), which block humanoid robots, sweeping robots, and wheelchair users; **door sill/floor height differences** (N = 82), which stop wheelchair users and sweeping robots; **narrow pathways** (N = 438), usually caused by narrow doors or furniture clutter, which can restrict humanoid robots, and also wheelchair users in extreme cases; **lacking of turning spaces** (N = 28), 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. The results are visualized in Figure 5.

5. Findings

Drawing on both quantitative results and qualitative inspection of model outputs, we distill three primary findings. Collectively, they indicate that current VLM-based navigation remains brittle under capability constraints, visually de-

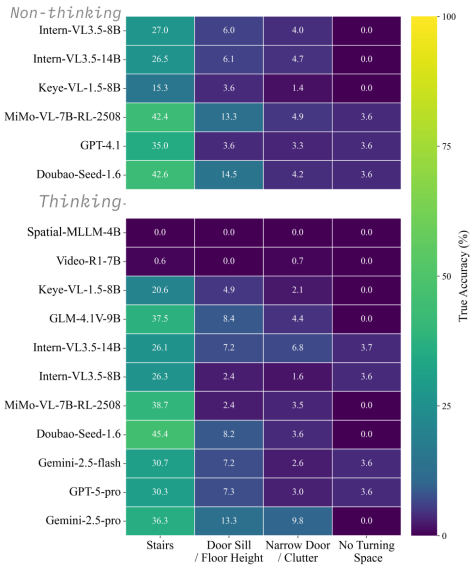

Figure 5. Model accuracy by obstacle type.

manding inputs, and geometry-sensitive traversability judgments. We also summarize actionable guidance for practitioners and outline concrete directions for future work.

5.1. Capability constraints induce systematic performance degradation

Across all tested models, performance is highest under the baseline embodiment without mobility limitations (adults without mobility constraints; mean CAPNAV score $= 57.83\%$). As constraints tighten, performance degrades. The largest gap appears for the HUMANOID embodiment, which diverges most from the HUMAN setting due to (i) an inability to climb stairs and (ii) a minimum pathway clearance requirement of $0.9\,\mathrm{m}$; this configuration attains the lowest average score (mean $= 39.12\%$).

This pattern highlights systematic deficits and imbalances in VLM-based navigation under mobility constraints. Strong performance in human-like settings does not translate to capability-constrained scenarios; hence, targeted evaluation is essential before VLM deployment in safety-critical navigation.

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, 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.

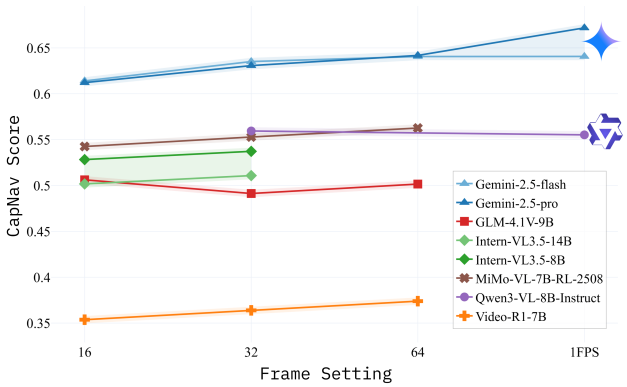

Figure 6. Performance across frame-budget settings. While higher frame counts can improve results, the gains are model-dependent and may saturate, consistent with a visual integration bottleneck.

5.2. Increasing visual budget helps, but exposes a visual bottleneck

Comparing matched inference settings with and without *thinking*, we find consistent improvements when *thinking* is enabled. Across 9 matched pairs, *thinking* configurations outperform their non-*thinking* counterparts by an average of ΔCAPNAV $= 6.87\%$. This gain is accompanied by a substantial compute cost: in our measurements, mean inference time increases by approximately $8\times$ (from $14.94\,\mathrm{s}$ to $123.94\,\mathrm{s}$).

Across the **16/32/64/1 FPS** frame settings, latency is comparatively stable, but higher frame counts increase GPU VRAM and API costs. Performance generally improves with more frames; however, the marginal benefit is uneven, tending to be larger for stronger models (notably Gemini) and smaller or negligible for weaker models (see Figure 6). This pattern reveals a *visual bottleneck*: additional evidence only helps if the model can reliably integrate it.

Qualitative inspection suggests two explanations: (i) For most open-sourced models, 64 frames is already a very heavy input but it can still remain sparse in large or complex scenes, requiring robust cross-frame spatial integration and viewpoint alignment that weaker models often lack. (ii) Additional viewpoints can increase *obstacle hallucinations* in some models: salient but non-blocking cues may be misinterpreted as traversal barriers, echoing prior observations about limitations in nonlocal reasoning for VLMs [4].

These results imply a cost–benefit frontier rather than a monotonic "more is better" rule. For CAPNAV-like VLN tasks, increasing inference budget can improve performance, but the return depends on model capacity and may trade off against latency, memory, cost, and even error modes.

5.3. Dimension neglect: models under-detect geometry-driven constraints

From our obstacle type analysis in Figure 5, all tested models perform markedly better on obstacles with salient visual signatures (e.g., stairs, door sills) than on constraints requiring implicit metric estimation (e.g., narrow clearances, turning radii). These failures indicate a recurring form of *dimension neglect*: traversability decisions that hinge on estimating size, clearance, or maneuvering space from images remain challenging for all tested VLMs.

To probe whether the above deficits are learnable, we ran a pilot fine-tuning experiment. We augment the CapNav dataset on the navigation graph to expand the task count from 2K to 13K, and LoRA fine-tune on Qwen3-VL-8B-Instruct. After one epoch of training, test score improves from 45.26% to 55.18%, and the finetuned model generates more dimension-aware reasoning. However, reasoning validity decreases ($0.30 \rightarrow 0.25$), and we observe frequent false positives on narrow passages (i.e., incorrectly flagging feasible routes as blocked).

These outcomes suggest that fine-tuning can reduce dimension neglect but does not fully resolve geometric precision, and may shift the error profile toward overly aggressive obstacle judgments. This highlights the need to jointly optimize for *obstacle awareness* and *dimension reasoning*, rather than treating score improvements alone as sufficient.

Building on this pilot, we plan to explore RL-based tuning with task-level rewards under embodiment constraints. We also aim to experiment with spatial-information injection (depth estimates, topological priors, or map-like intermediate representations) to improve spatial dimension judgments.

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 evaluating 13 state-of-the-art VLMs, we observe three consistent patterns. First, performance degrades systematically as mobility constraints tighten, indicating that strong results in unconstrained human-like settings do not transfer reliably to other embodiments. Second, increasing the inference budget (more frames or "thinking" mode) yields inconsistent gains and incurs computational cost, exposing a visual integration bottleneck. Third, models exhibit dimension neglect: they handle salient obstacles better but struggle with dimension reasoning (*e.g.*, narrow clearances and turning radii). A pilot fine-tuning study suggests that these deficits are partially learnable, but fine-tuning may introduce false positives, highlighting the need to jointly optimize the spatial reasoning reliability in task formulation and spatial dimension. We release both the CapNav dataset and annotation tool.

References

[1] Peter Anderson and et al. On evaluation of embodied navigation agents. In *arXiv:1807.06757*, 2018. 2

[2] Peter Anderson and et al. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, 2018. 2, 3, 6

[3] 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

[4] Shmuel Berman and Jia Deng. Vlms have tunnel vision: Evaluating nonlocal visual reasoning in leading vlms. *arXiv preprint arXiv:2507.13361*, 2025. 8

[5] 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

[6] 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. 4

[7] 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

[8] Angel Chang and et al. Matterport3d: Learning from rgb-d data in indoor environments. In *3DV*, 2017. 2, 5

[9] 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

[10] 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

[11] An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatial-rgpt: Grounded spatial reasoning in vision-language models. *Advances in Neural Information Processing Systems*, 37:135062–135093, 2024. 2

[12] 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. 7

[13] 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

[14] 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. 2, 6, 7

[15] 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

[16] 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. 2

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

[18] Meera Hahn, Devendra Singh Chaplot, Shubham Tulsiani, Mustafa Mukadam, James M Rehg, and Abhinav Gupta. No rl, no simulation: Learning to navigate without navigating. *Advances in Neural Information Processing Systems*, 34:26661–26673, 2021. 3

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

[20] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. *arXiv preprint arXiv:2210.05714*, 2022. 2

[21] Gabriel Ilharco and et al. General evaluation for instruction conditioned navigation using dynamic time warping. *arXiv:1907.05446*, 2019. 2

[22] 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. 2, 3

[23] 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

[24] 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

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

[26] 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

[27] 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

[28] 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

[29] 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

[30] 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. 2

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

[32] 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

[33] 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

[34] 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

[35] 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

[36] 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

[37] OpenAI. Introducing gpt-4.1 in the api. https://openai.com/index/gpt-4-1/, 2025. Accessed: 2025-11-12. 7

[38] OpenAI. Introducing gpt-5. https://openai.com/index/introducing-gpt-5/, 2025. Accessed: 2025-11-12. 7

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

[40] Yanyuan Qiao, Haodong Hong, Wenqi Lyu, Dong An, Siqi Zhang, Yutong Xie, Xinyu Wang, and Qi Wu. Navbench: Probing multimodal large language models for embodied navigation. *arXiv preprint arXiv:2506.01031*, 2025. 3

[41] Santhosh K. Ramakrishnan and et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv:2109.08238*, 2021. 2, 5

[42] Santhosh Kumar Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. Poni: Potential functions for objectgoal navigation with interaction-free learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18890–18900, 2022. 3

[43] 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

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

[45] ByteDance Seed. Seed 1.6. https://seed.bytedance.com/en/seed1_6/, 2025. Accessed: 2025-11-12. 7

[46] Dhruv Shah and et al. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In *CoRL (PMLR)*, 2023. 2

[47] 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. 2

[48] 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

[49] 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. 2

[50] 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. 7

[51] Qwen Team. Qwen3 technical report, 2025. 7

[52] 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. 7

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

[54] Liuyi Wang, Xinyuan Xia, Hui Zhao, Hanqing Wang, Tai Wang, Yilun Chen, Chengju Liu, Qijun Chen, and Jiangmiao Pang. Rethinking the embodied gap in vision-and-language navigation: A holistic study of physical and visual disparities. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9455–9465, 2025. 2, 3

[55] 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

[56] 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

[57] 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. 2, 6, 7

[58] 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

[59] 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.

[60] 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

[61] Peiran Xu, Sudong Wang, Yao Zhu, Jianing Li, and Yunjian Zhang. Spatialbench: Benchmarking multimodal large language models for spatial cognition. *arXiv preprint arXiv:2511.21471*, 2025. 2

[62] 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. 7

[63] 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. 2

[64] Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, et al. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. *arXiv preprint arXiv:2502.09560*, 2025. 3

[65] 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

[66] Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. *arXiv:2305.16986*, 2023. 2

[67] 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

[68] 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. 7

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

Supplementary Material

Overview of Supplementary Material

This supplementary material provides examples of VLM input and output in the dataset generation and benchmark evaluation process. We include the following components: (1) the exact prompts and input representations provided to VLMs, (2) examples of ground truth

A. Task Generation

A.1 Prompt Template

We use a unified prompt template to instruct a VLM to generate navigation tasks for each scene. The template contains three components: (1) an overall instruction, (2) file attachments, *i.e*. a scene video and a guideline document that contains more detailed instructions (Sec A.2-3), and (3) a list of spatial nodes in the scene (Sec A.4).

An example of generated tasks can be found in `taskGeneration/HM3D00025_tasks.json`.

Instruction:

You are an expert benchmark question designer for the Capability-Conditioned Navigation Benchmark. You are provided with three materials:

1. A video showing the indoor environment.
2. A guideline PDF defining how to create navigation questions.
3. A list of valid scene nodes.

Requirements:

Please strictly follow the guideline and generate realistic, visually grounded, route-based navigation questions. Each question must:

- Include a start and end node with detailed in-room descriptions.
- Follow the schema structure provided by the system.
- Output a valid JSON array (no extra commentary).
- Ensure the total number of generated questions is $\geq$ {min_questions}.

External files (video, pdf)

Node List:

{node_list_text}

A.2 Video Input

When generating navigation tasks, the VLM receives a touring video of the space. In Fig.7, we show representative frames of an example video. The full MP4 file is included in the supplementary ZIP, see `taskGeneration/HM3D00025.mp4`.

Figure 7. Representative frames from the input video `HM3D00025.mp4`. The VLM receives the complete MP4 video for navigation task generation.

A.3 Guideline Document

We provide a carefully designed guideline document to the VLM to specify the exact type and format of the navigation tasks to be generated. We show an excerpt of the document below. The complete guideline is included in the supplementary ZIP (`taskGeneration/Generate_Tasks_Guidelines.pdf`).

Guideline (excerpt):

```
Capability-conditioned navigability

1. Goal
This guideline defines how to generate
capability-neutral, route-based
navigation questions for the
Capability-Conditioned Navigation
Benchmark. You are an AI model that
generates route-based navigation
questions from an indoor walkthrough
video.
Your task is to create realistic and
scene-consistent navigation tasks that
test how [Agent] can move and act
within the visible environment.

Each question must:
1. Be fully grounded in the video
(only describe objects, rooms, and
connections that actually appear).
```

```
2. Use the provided list of scene
nodes as the only valid room-level
landmarks.
...
```

Note: Full guideline PDF in the supplementary ZIP.

A.4 Node List

To ground the sub-spaces in a scene, the VLM also receives a list of node IDs paired with textual descriptions. We instruct the VLM to reference these nodes when generating navigation tasks. Below we show a few nodes from the scene `HM3D00025`, taken directly from the scene graph. The full JSON file is included in the supplementary ZIP (`groundTruth/HM3D00025-graph.json`).

Node List:

```
{
node_117: Master bedroom foyer
node_118: Master bedroom bathroom
node_119: Master bedroom home office
node_120: Top floor stair landing and
hallway
...
node_131: First floor bedroom with
twin beds
node_132: First floor bathroom
}
```

Note: We show a subset of the nodes here. The model receives the full node list during task generation.

B. Benchmark Evaluation

When evaluating on the CapNav benchmark, we query VLMs to answer the capability-conditioned navigation questions. Each query provides the following inputs:

1. a specific indoor scene shown as a tour video (see Figure 7),
2. all nodes of the scene's navigation graph,
3. one navigation question,
4. one agent profile describing the mobility capabilities.

The model must use both the video and the text inputs to determine whether the agent can complete the navigation task. Below we show a prompt example for scene `HM3D00025` and agent HUMANOID. The original prompt file can also be seen in `benchmarkEvaluation/HM3D00025_q01_HUMANOID.txt`

B.1 Example Prompt (HM3D00025, HUMANOID)

Instruction:

You are an expert visual reasoning agent for indoor navigation tasks. You will receive four materials:

1. A video showing the indoor environment.
2. A single navigation question asking whether the agent can move from one area (node) to another.
3. The agent's physical and capability profile.
4. The list of all nodes in this environment, with their textual descriptions (e.g., room type, furniture, width of passages).

Your goal is to determine whether the agent can successfully reach the destination area from the starting area, based on both the video and the textual descriptions. You must reason step by step about spatial constraints, obstacles, connectivity, and the agent's mobility limitations.

Input:

1.Navigation Question:

Can [Agent] move from the dark sectional sofa area in the main living space to the white L-shaped sofa area in the first floor living room?

2.Agent Profile (HUMANOID):

```
Agent name: HUMANOID
Body shape: box
Height (m): 1.5
Width (m): 0.9
...
Can operate elevator: True
Can open the door: True
Description: Boston Dynamics Atlas
humanoid robot approximately the size of
a human, capable of obstacle crossing up
to 0.4 m, for a single obstacle.
However, it cannot go up stairs.
```

3.Scene Node List:

```
node_117 - Master bedroom foyer
node_118 - Master bedroom bathroom
node_119 - Master bedroom home office
...
node_131 - First floor bedroom with twin
beds
node_132 - First floor bathroom
```

4.Video

(You can observe the video for spatial layout and obstacles.)

Task:

Your goal is to determine whether the agent can **navigate** from the start area to the goal area.
Focus exclusively on **movement feasibility**, considering physical dimensions, obstacle heights, and connection constraints.
You must **not stop after a single failed route attempt**. If one possible route is blocked (e.g., by stairs or narrow spaces), you must **actively consider all other possible paths** between the start and goal nodes in the scene graph.

Follow these principles:

1. Explore **all possible routes** through the scene graph before deciding the task is impossible. That means, try **multiple alternative routes** using all visible connections in the scene graph until you are confident that **no feasible route** exists.
2. Account for the agent's capabilities (e.g., door opening, elevator operation, stair traversal) when evaluating possible paths.
3. When multiple feasible paths exist, select the **most direct and realistic one** given the agent's capabilities.
4. If no route works, specify which **edge or physical barrier** prevents traversal and explain why.
5. If a feasible route exists, specify the **sequence of nodes** representing the navigable path.

Important:
If you initially find the route impossible, **re-examine the scene graph** and attempt at least two distinct alternative paths before concluding "no".
Your reasoning should reflect persistent exploration: do not assume failure after one obstacle; explore until all logical alternatives are ruled out.
You do not need to consider unrelated interactions (e.g., turning on lights, using computers, or touching furniture).

Please decide for each given question whether the agent can complete the navigation task.
If yes, provide a **feasible path** through the relevant nodes.
If no, specify the **edge (two connected nodes)** that blocks traversal and give a concise **reason** (e.g., too narrow passage, stairs, or closed door).
Your reasoning should always consider the agent's physical capabilities mentioned in the Agent profile(e.g., wheelchair cannot climb stairs, sweeper robot cannot open doors).

Return your answer in the required structured JSON format below.

Output Format (JSON only):

```
[
    ...
    {
        "question": "...",
        "agent": "HUMANOID",
        "result": {
            "answer": "no",
            "path": ["node_12",
            "node_14", "node_15"],
            "reason": "Too narrow
            passage between the sofa
            and wall"
        }
    }
]
```

Return only the JSON array, no explanation, commentary, or markdown formatting.

C. Reasoning Evaluation

CapNav deploys an LLM-as-judge method to evaluate whether a navigability explanation from a VLM's output is consistent with the ground-truth annotations. We provide the LLM judge with (1) the VLM-generated reasoning for why a path is non-traversable and (2) the full ground-truth traversability record for the path. The LLM will determine whether the explanation correctly captures the underlying failure conditions.

Below we briefly present the prompts and data formats. For a full example, please check `reasoningEvaluation / HM3D00025 _ q01 _ HUMANOID.txt`

C.1 Prompt Template

Instruction (Reasoning Evaluation):

```
You are evaluating whether a
navigation system's reasoning is
correct.

You are given:
1. A system-generated explanation of
why a path is not traversable.
2. The ground-truth traversability
data for all edges along the path.

Your task:
- Determine if the system's reasoning
correctly identifies or aligns with
the actual traversability issues.
```

```
- Exact wording is not required;
conceptual correctness is sufficient.
- Grant partial credit if the
reasoning identifies some but not all
issues.

Respond in JSON format with:
{
   "correct": true/false,
   "explanation": "Brief explanation
   of why the reasoning is correct or
   incorrect"
}
```

C.2 Example Input (Excerpt)

Example Input (excerpt):

```
System reasoning:
"The agent cannot traverse stairs, and
the only path between the
Basement bar space (node_109) and the
Laundry room (node_105)
requires going up the Basement stairs
(node_114 or node_115),
which the agent cannot do."

Ground-truth traversability (excerpt):
{
  "total_edges": 6,
  "traversable": 4,
  "non_traversable": 2,
  "edges": [
    {
      "from": "node_109",
      "to": "node_108",
      "from_name": "Basement bar
      space",
      "to_name": "Pool and media
      room",
      "exists": true,
      "traversable": true,
      "note": "Edge from \"Basement
      bar space\" to \"Pool and media
      room\"",
      "ground_truth": {
        "exists": true,
        "traversable": true,
        "note": "Edge from \"Basement
        bar space\" to \"Pool and
        media room\""
      }
    },
    ...
  ]
}
```