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

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.1, we show representative frames of an example video. The full MP4 file is included in the supplementary ZIP, see `taskGeneration/HM3D00025.mp4`.

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 supple-

Figure 1. Representative frames from the input video `HM3D00025.mp4`. The VLM receives the complete MP4 video for navigation task generation.

mentary 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 1),
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\""
      }
    },
    ...
  ]
}
```