
NavBench: Probing Multimodal Large Language Models for Embodied Navigation - Supplementary

Yanyuan Qiao¹ Haodong Hong²³ Wenqi Lyu⁴ Dong An⁵ Siqi Zhang⁶
Yutong Xie⁵ Xinyu Wang⁴ Qi Wu^{4*}

¹Swiss Federal Institute of Technology Lausanne (EPFL)

²The University of Queensland ³CSIRO Data61 ⁴The University of Adelaide

⁵Mohamed bin Zayed University of Artificial Intelligence ⁶Tongji University

A Model Details

We list the full model names corresponding to the shorthand used in the main text, along with their sources and access methods, as illustrated in Table 1:

Table 1: Full model names used in evaluation.

Shorthand	Full Name
GPT-4o	gpt-4o-2024-08-06
GPT-4o-mini	gpt-4o-mini-2024-07-18
InternVL2.5-2B	InternVL2.5-2B
InternVL2.5-8B	InternVL2.5-8B
Qwen2.5-VL-3B	Qwen2.5-VL-3B-Instruct
Qwen2.5-VL-7B	Qwen2.5-VL-7B-Instruct
LLaVA-OneVision-7B	llava-onevision-qwen2-7b-ov
Llama3.2-Vision-11B	Llama3.2-Vision-Instruct

B Details of Dataset Construction

Hyperparameter Weights Used in Complexity Scoring The weights for each complexity formula were empirically chosen based on feature distributions across a random subset of 50 scenes, ensuring that all components contributed comparably to the final score. Specifically, we used:

- $\alpha_1 = 0.5, \alpha_2 = 1.0, \alpha_3 = 2.5, \alpha_4 = 0.6$
- $\beta_1 = 0.7, \beta_2 = 0.6, \beta_3 = 0.5, \beta_4 = 0.5, \beta_5 = 1.0$
- $\gamma_1 = 2.5, \gamma_2 = 0.4, \gamma_3 = 2.8, \gamma_4 = 2.0$

To evaluate the robustness of our complexity scoring design, we conducted a sensitivity analysis by independently perturbing each weight by $\pm 20\%$ while keeping the others fixed. For each perturbed configuration, we re-scored all 432 cases and re-assigned difficulty levels. We then computed the agreement rate with the original classification. As shown in Figure 1, most weights led to minimal changes in categorization, with agreement rates typically around 90%. A few parameters, such as those associated with execution complexity (*e.g.*, γ_1, γ_3), showed higher sensitivity, which reflects their stronger influence on the final score. These results confirm that the scoring framework is generally robust to moderate parameter variations while remaining sensitive to semantically impactful dimensions.

*Corresponding author

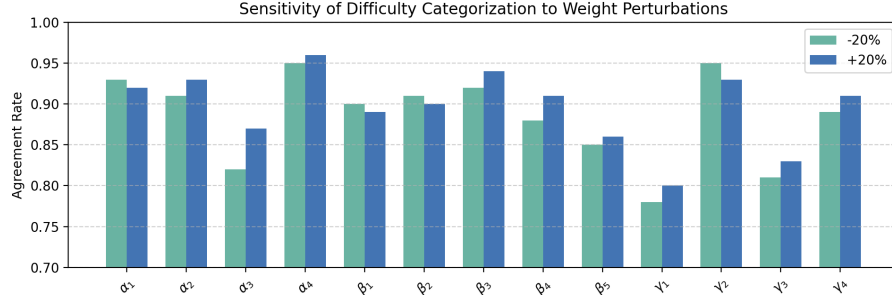


Figure 1: Sensitivity analysis of complexity-based difficulty categorization. Each bar shows agreement rate after applying a $\pm 20\%$ perturbation to the corresponding weight while keeping others fixed.

Navigation Episodes Difficulty Rating

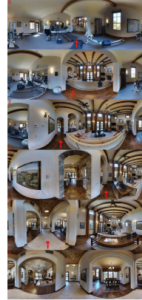
Below is a navigation episode, including the instruction and visualized trajectory. Please rate the difficulty based on your judgment of how challenging it would be for an agent to follow the instruction and complete the navigation in this environment.

You will rate the task along three independent dimensions:

- **Cognitive Complexity** – How difficult is the instruction to understand and interpret? Consider instruction length, action verbs, spatial terms, and landmark references.
- **Spatial Complexity** – How complex is the physical layout of the environment? Consider distance, spatial coverage, and path geometry.
- **Execution Complexity** – How much effort is required to carry out the navigation? Consider number of steps, turns, and transitions.

Please use a scale from 1 (very easy) to 9 (very hard). There are no right or wrong answers, use your best judgment for each dimension.

- ★ 1. **Instruction:** Go straight and exit the room using the arch ahead. Turn left and go straight past the couches. Turn right and then turn left into the second arch. Wait near the kitchen. (Distance to navigate: 11.7 meters)



	1	2	3	4	5	6	7	8	9
Cognitive Complexity	★	★	★	★	★	★	★	★	★
Spatial Complexity	★	★	★	★	★	★	★	★	★
Execution Complexity	★	★	★	★	★	★	★	★	★

Figure 2: Screenshot of the human evaluation interface used for annotator scoring. Annotators rated each case across spatial, cognitive, and execution dimensions based on structured guidelines.

Details of Human Evaluation To validate the difficulty scores produced by our automated scoring system, we conducted a human evaluation study. Each of the navigation cases was independently rated by five annotators along three dimensions: *Cognitive Complexity*, *Spatial Complexity*, and *Execution Complexity*, using a 9-point scale. Annotators were shown the natural language instruction, physical navigation distance, and a visualized panoramic trajectory.

Structured guidelines were provided at the top of the interface to ensure consistent interpretation of each dimension. A screenshot of the evaluation form is shown in Figure 2. The five annotators were

all graduate students familiar with embodied navigation tasks, with backgrounds in computer vision or robotics research.

C Quality Control for Temporal Progress Estimation Data

The QA pairs for the Temporal Progress Estimation task were selected from a larger candidate pool through a combination of automatic filtering and manual review. We first applied an automatic rule to exclude QA pairs where two adjacent sub-instructions ended at the same viewpoint, as such cases typically indicate unclear temporal boundaries. After removing these structurally ambiguous pairs, we manually reviewed the remaining candidates and excluded samples that lacked a clear or meaningful sense of temporal progress, such as those with minimal spatial change between chunk images or ambiguous sub-instruction-to-trajectory alignment. This two-stage filtering process ensured that the final QA pairs exhibit visually and semantically distinct progression steps.

D Calibration of Complexity Score Weights

To determine the weights (α, β, γ) in the complexity scoring framework, we conducted a detailed statistical analysis of the numerical features associated with each complexity dimension: spatial, cognitive, and execution.

First, we extracted structured features from each trajectory. Spatial complexity included path length, standard deviation of turning angles, elevation range, and two-dimensional area coverage. Cognitive complexity was based on instruction length, number of verbs, number of spatial terms, number of landmark mentions, and number of subordinate clauses. Execution complexity was measured by the number of steps, number of turns, presence of floor changes, and number of decision points.

Second, we sampled data from 50 scenes and computed summary statistics for each feature, including mean, standard deviation, and min–max range. This analysis helped us understand the scale and variation of each feature. For example, instruction length exhibited a broader numerical spread than the number of landmark mentions, which would otherwise lead to a disproportionate impact on the cognitive complexity score.

Third, we initialized all feature weights equally within each complexity category and then adjusted them based on the observed distributions to ensure balanced contributions. Features with inherently larger value ranges were assigned smaller weights to prevent them from dominating the overall score. The aggregation weights across the three dimensions (α, β, γ) were then tuned to align the automatic scores with the distribution of human difficulty ratings.

Finally, we evaluated the robustness of the weighting scheme through a sensitivity analysis, perturbing each weight by 20% while keeping the others fixed. As shown in Figure ??, over 90% of samples retained their original difficulty category, indicating that the scoring method is stable with respect to weight variation.

E Details of Real-World Deployment

To demonstrate the implementation of our real-world navigation pipeline, we describe its full processing flow as follows.

The robot’s onboard RGB camera captures images from 12 discrete directions at 30-degree intervals. These are resized to $224 \times 224 \times 3$ and combined into a panoramic observation. To obtain geometric information, we integrate a monocular depth estimation model [1] that produces scale-consistent depth maps at 256×256 resolution for indoor scenes.

Given the processed RGB and depth observations, the Waypoint Predictor extracts features using two independent ResNet-50 backbones. Let I_i^{rgb} and I_i^{d} denote the RGB and depth images at direction i , and the fused feature is computed as:

$$v_i^{\text{rgbd}} = W_m(f_{\text{rgb}}(I_i^{\text{rgb}}) \parallel f_{\text{depth}}(I_i^{\text{d}})). \quad (1)$$

Table 2: VLN-Bench (tiny) Evaluation Results

Model	Navigation Comprehension				Navigation Execution							
	Global	Progress	Local	Comp. Avg	Easy		Medium		Hard		Exec. Avg	
	Accuracy				SR	SPL	SR	SPL	SR	SPL		
Chance Level (Random)	19.17	19.00	31.00	23.73	14.58	10.15	7.33	3.54	8.33	5.56		8.08
Human Level	88.33	79.00	85.00	84.11	91.67	88.68	87.50	81.53	75.00	65.17		81.59
<i>Closed Models</i>												
GPT-4o	51.67	45.00	63.00	53.89	66.08	49.01	43.79	36.44	25.00	20.11		40.07
GPT-4o-mini	49.17	32.00	57.00	46.06	43.83	37.62	29.17	25.96	16.67	10.76		27.67
<i>Open-Source Models</i>												
InternVL2.5-2B	71.67	22.00	13.00	35.56	26.54	23.37	10.42	8.50	8.33	6.77		13.99
Qwen2.5-VL-3B	62.50	22.00	48.00	44.83	28.71	22.12	12.00	9.08	8.33	7.90		14.69
InternVL2.5-8B	60.00	32.00	26.00	39.33	26.79	25.37	13.75	10.67	8.33	7.90		15.47
Qwen2.5-VL-7B	36.67	32.00	47.00	38.56	46.25	35.59	25.27	18.93	12.50	5.93		24.41
LLaVA-OneVision-7B	30.83	28.00	41.00	33.28	32.42	19.99	15.08	9.52	14.25	7.29		16.76
Llama3.2-Vision-11B	36.67	24.00	27.00	29.89	29.17	25.44	12.50	11.33	11.04	8.81		16.38

These fused features are passed to a lightweight transformer that models spatial relationships between the current and adjacent views. The resulting output is transformed into a spatial heatmap and refined using non-maximum suppression (NMS) to obtain the top- K salient waypoints:

$$H_{\text{refined}} = \text{NMS}(\text{MLP}(v_i^{\text{rgbd}})). \quad (2)$$

Each selected waypoint is represented by a relative angle and distance from the agent’s current location and is forwarded to the MLLM Decision Module. The MLLM receives the navigation instruction and the corresponding RGB views of the K candidates, selecting the most goal-aligned waypoint through multimodal reasoning.

Finally, the Low-Level Controller converts the chosen angle and distance into executable rotation and translation commands, which are issued to the robot’s chassis API for physical execution.

F VLN-Bench (tiny) Human Performance

To establish a human performance reference for VLN-Bench, we construct a compact evaluation subset named VLN-Bench (tiny) by randomly sampling a representative subset of tasks, inspired by the strategy in [2]. Annotating the entire dataset with human responses is impractical due to scale, so we select 120 multiple-choice questions for the Global Instruction Alignment task, 100 for Temporal Progress Estimation, 100 for Local Observation-Action Reasoning, and 72 navigation episodes for Execution Evaluation. All questions are in multiple-choice format with predefined correct answers. Human participants, who were volunteer students from our research institute with relevant backgrounds, independently completed each task. Their responses were automatically scored using the same metrics applied to model evaluation, including accuracy for comprehension tasks and SR/SPL for execution. This evaluation did not involve personal data collection or behavioral intervention and was conducted in accordance with institutional guidelines. In total, VLN-Bench (tiny) includes 392 human-evaluated questions and episodes. The results, shown in Table 2, serve as an empirical reference point for model performance on VLN-Bench (tiny).

G Multilingual Evaluation

To explore the generalization of models across languages, we conducted a preliminary multilingual study on the Global Instruction Alignment task, a component of the navigation comprehension evaluation. This task involves aligning a textual instruction with a visual trajectory and is particularly suited for analyzing the effect of instruction language, as it directly measures the model’s ability to semantically match linguistic and visual information.

Two representative models were selected: GPT-4o, the strongest closed-source model, and Qwen2.5-VL-7B, the best-performing open-source model. Thirty instructions were randomly sampled and translated into four languages. Italian and German were chosen as typologically close to English, while Hindi and Telugu were included as linguistically distant languages that also appear in the multilingual RxR dataset. To isolate the effect of instruction language, all other prompt components

were kept in English. Translations were performed automatically without human post-editing to simulate a zero-shot cross-lingual setting.

Table 3: Multilingual Global Instruction Alignment Accuracy

Model	English	Hindi	Telugu	Italian	German
GPT-4o	55.00	37.50	49.17	45.00	51.67
Qwen2.5-VL-7B	62.50	48.33	32.50	60.83	54.17

Performance degradation is relatively minor for Italian and German, whereas both models show noticeable drops for Hindi and Telugu. These results suggest that language differences can significantly affect model performance, particularly when the instruction language diverges from English. Since this pilot study was based on automatic translation without human post-editing, the results are considered exploratory. Future work will extend this evaluation to a larger multilingual dataset and investigate the use of language identifier tokens to enhance cross-lingual robustness.

H Limitations and Future Work

NavBench currently focuses on indoor navigation tasks, which are foundational to many embodied AI applications. However, real-world scenarios, such as those encountered in household robotics, often require agents to perform navigation in conjunction with object manipulation, such as locating, retrieving, or interacting with items in the environment. While NavBench provides a foundation for evaluating navigational reasoning, it does not yet support the assessment of such integrated capabilities.

In future work, we plan to extend NavBench to encompass joint navigation and manipulation tasks. This would allow for a more comprehensive evaluation of embodied intelligence, bridging perception, language understanding, spatial reasoning, and physical interaction in a unified framework.

I Details of Error Analysis

We analyzed 100 failed episodes sampled from both proprietary and open-source models, covering all difficulty levels in NavBench. Each case includes the executed trajectory, the model-generated reasoning trace (“thought”), and the next-step plan, allowing us to assess not only the agent’s actions but also its underlying decision-making process.

For each failure, we first recorded detailed notes on the suspected causes of error. These annotations were based on observed inconsistencies between the instruction, the plan, and the resulting actions. After reviewing all annotated cases, we grouped the failures into four major categories, as reported in the main paper.

For example, in one case with the instruction “*walk down the stairs and stop on the landing*”, the agent successfully reached the target location but mistakenly believed it was still en route, stating “*I need to continue to go down the stairs.*” As a result, it continued navigating unnecessarily, reflecting a misunderstanding of task progress. This error highlights the model’s difficulty with temporal estimation and corresponds to its poor performance on the Temporal Progress Estimation subtask.

J Prompt Templates

We provide detailed prompt templates for each sub-task within NavBench. These templates are designed to clearly define the objectives and input-output requirements for the models, ensuring consistency and reproducibility across evaluations. Each template includes specific instructions, input formats (e.g., panoramic views, navigation trajectories, and candidate options), and expected output structures tailored to the corresponding sub-task, as illustrated in Figures 3.

K NavBench Examples

In Figure 4 to Figure 7, we provide examples from NavBench’s comprehension tasks, including Global Instruction Alignment, Temporal Progress Estimation, and Local Observation-Action Reasoning. These visualizations illustrate the structure and format of the input scenes, questions, and answer choices. We omit execution task examples due to their dependence on interactive simulator dynamics, which are less amenable to static visualization.

References

- [1] S. Bhat, Reiner Birkel, Diana Wofk, Peter Wonka, and Matthias Muller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *ArXiv*, abs/2302.12288, 2023.
- [2] Jihan Yang, Shusheng Yang, Anjali Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in Space: How Multimodal Large Language Models See, Remember and Recall Spaces. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Prompt Template for Global Instruction Alignment

You are presented with a sequence of 360-degree panoramic views that represent a navigation path from the starting point to the goal location. Each panorama has a red arrow indicating the direction of movement and a red step number in the top left corner. The final panorama shows the goal location. You are also provided with five different instructions, but only one accurately describes the complete path. Identify the correct instruction based on the panoramas. Only return the index of the correct instruction like 'Instruction X'.

Trajectory Views: {navigation trajectories}

Instructions: {Instructions}

Prompt Template for Temporal Progress Estimation

You are given a navigation instruction divided into multiple sub-instructions, along with a series of 360-degree panoramic views depicting the path taken so far from the starting point to the current, incomplete segment of the overall path described by the full instruction. Your task is to determine how many sub-instructions have been completed based on the views provided. Only return the index of the last completed sub-instruction like 'Sub-instruction X'.

Trajectory Views: {trajectory views}

Sub-instructions: {sub-instructions}

Prompt Template for Future-Action Prediction

You are given two panoramas taken from two nearby locations in the same environment. The first panorama represents the current location, and the second panorama represents a nearby location. You are also given a set of candidate locations, each represented by a single view indicating the moving directions at the current location. Your task is to identify the candidate location that corresponds to the second panorama. Only return the index of the correct candidate location like 'Candidate X'.

Current location: {current location}

Candidate views: {candidate views}

Prompt Template for Future-Observation Prediction

You are given a panoramic image taken from a specific location within an environment, alongside a single image indicating the moving direction. Additionally, you have a set of candidate locations, each represented by a panoramic image taken at that location. Your task is to identify the candidate location that matches the direction provided. Only return the index of the correct candidate location like 'Candidate X'.

Current location: {current location}

Moving direction: {candidate locations}

Figure 3: Prompt templates used in VLN-Bench, covering four core tasks: Global Instruction Alignment, Temporal Progress Estimation, and two sub-tasks for Local Observation-Action Reasoning (Future-Action Prediction and Future-Observation Prediction).

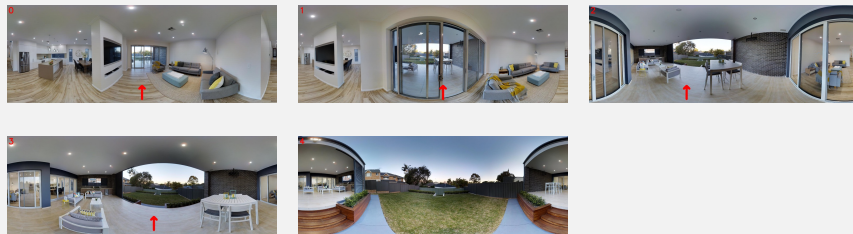
Global Instruction Alignment



Question:

You are presented with a sequence of panoramic views that represent a navigation path from the starting point to the goal location. Identify the correct instruction.

- A. Enter the bottom floor through the rose colored carpet on the left. Make a hard left just before the map picture. Make a right opposite a familyroom on the far wall to the left. Walk past the windows on the left. Wait in the threshold of the small corner to the left of the wooden beam.
- B. Enter the tan door through gym on the left. Make a hard left just before the tile mosaic. Make a right opposite on pool platform on the large table to the left. Walk past the right and large windows on the left. Wait in the threshold of potted plant to the left of the leftmost archway.
- C. Enter two book stands through the red carpet on the left. Make a hard left just before the stools. Make a right opposite the corridor on stair landing to the left. Walk past the first set of stairs on the left. Wait in the threshold of the light switch to the left of the wooden chair.
- D. Enter the purple sofa through another room on the left. Make a hard left just before a double sink. Make a right opposite cross on library area on the left to the left. Walk past 3 more steps on the left. Wait in the threshold of library area to the left of a big starburst tile.
- E. Enter the house through the open french doors on the left. Make a hard left just before the bed. Make a right opposite the thermostat on the wall to the left. Walk past the console table on the left. Wait in the threshold of the door to the left of the linen cupboard.



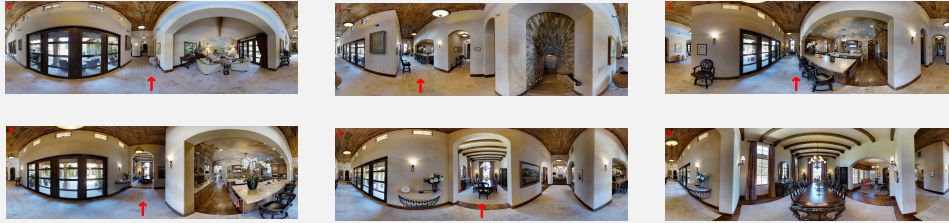
Question:

You are presented with a sequence of panoramic views that represent a navigation path from the starting point to the goal location. Identify the correct instruction.

- A. Go down the walkway between the bar and the floor cabinets. Make a left at the door leading to the bedroom.
- B. take the stairs then head for the kitchen. stop at the bottom of the stairs.
- C. walk towards yellow chair and turn right, walk past couch and straight through doorway towards dining table, walk on left side of dining table through double doors, walk over one doormat, stop on the second doormat.
- D. Walk outside of the patio doors. Go down the stairs to the grass area and then stop.
- E. Follow the pink carpet out of the room you are in, and through the next room. Exit the doorway of the next room and stop.

Figure 4: Examples of NavBench (Part 1).

Global Instruction Alignment



Question:

You are presented with a sequence of panoramic views that represent a navigation path from the starting point to the goal location. Identify the correct instruction.

- A. and stop, walk to the end of the entrance way, and turn left, turn put the exit of the building on your left, and wait one meter from the closest end of the long dining table, travel across the kitchen area with the counter and chair on your right, enter the room, continue straight until you reach the dining room
- B. walk to the end of the entrance way, travel across the kitchen area with the counter and chair on your right, continue straight until you reach the dining room, and stop, and turn left, enter the room, turn put the exit of the building on your left, and wait one meter from the closest end of the long dining table
- C. turn put the exit of the building on your left walk to the end of the entrance way and turn left travel across the kitchen area with the counter and chair on your right continue straight until you reach the dining room enter the room and stop and wait one meter from the closest end of the long dining table
- D. and stop, turn put the exit of the building on your left, and turn left, walk to the end of the entrance way, travel across the kitchen area with the counter and chair on your right, enter the room, and wait one meter from the closest end of the long dining table, continue straight until you reach the dining room
- E. enter the room, travel across the kitchen area with the counter and chair on your right, continue straight until you reach the dining room, turn put the exit of the building on your left, and turn left, and stop, and wait one meter from the closest end of the long dining table, walk to the end of the entrance way



Question:

You are presented with a sequence of panoramic views that represent a navigation path from the starting point to the goal location. Identify the correct instruction.

- A. Walk on into the kitchen and turn to the north. Walk past the staircase, behind the chairs. Walk to the north of the pillar. Stop and wait by the footstool.
- B. Walk on into the kitchen and turn to the forward. Walk past the staircase, behind the chairs. Walk to the forward of the pillar. Stop and wait by the footstool.
- C. Walk on into the kitchen and turn to the south. Walk past the staircase, behind the chairs. Walk to the south of the pillar. Stop and wait by the footstool.
- D. Walk on into the kitchen and turn to the right. Walk past the staircase, behind the chairs. Walk to the right of the pillar. Stop and wait by the footstool.
- E. Walk on into the kitchen and turn to the left. Walk past the staircase, behind the chairs. Walk to the left of the pillar. Stop and wait by the footstool.

Figure 5: Examples of NavBench (Part 2).

Temporal Progress Estimation



Question:

You are given a navigation instruction divided into multiple sub-instructions, along with a trajectory. Your task is to determine how many sub-instructions have been completed based on the views provided.

Sub-instruction 1: turn to your left

Sub-instruction 2: exit the room out of the door beside the wooden drawer

Sub-instruction 3: once out of the room walk across the small area and through the next entry way on the left

Sub-instruction 4: stop inside the room before you get to the door lead outside



Question:

You are given a navigation instruction divided into multiple sub-instructions, along with a trajectory. Your task is to determine how many sub-instructions have been completed based on the views provided.

Sub-instruction 1: exit the closet

Sub-instruction 2: walk past the drapery on the right

Sub-instruction 3: wait at the threshold of the bedroom door



Question:

You are given a navigation instruction divided into multiple sub-instructions, along with a trajectory. Your task is to determine how many sub-instructions have been completed based on the views provided.

Sub-instruction 1: enter the room directly in front of the stair.

Sub-instruction 2: go into the closet

Sub-instruction 3: continue into the bathroom

Sub-instruction 4: stop in front of the sink

Figure 6: Examples of NavBench (Part 3).

Local Observation-Action Reasoning

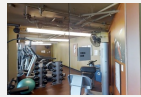


Current view



Target view

Question: Given the current view and a target view, select the direction that is most likely to lead to the target view.



A



B



C



D



Current view



Direction

Question: Given the current view and a target view, select the direction that is most likely to lead to the target view.



A



B



C



D



E



Current view



Direction

Question: Given the current view and a direction to move, select the location matching the expected view after moving.



A



B



C



D



Current view



Direction

Question: Given the current view and a direction to move, select the location matching the expected view after moving.



A



B



C



D

Figure 7: Examples of NavBench (Part 4).