

Are Multimodal Large Language Models Ready for Omnidirectional Spatial Reasoning?

Supplementary Material

Appendix

A. Algorithm Detail

In this section, we provide more details on the construction pipeline for our *OSR-Bench*.

A.1. QA Pair Generation

Algorithm 2 details our systematic approach to generating question-answer pairs from omni-cognitive maps. For each scene, we generate three types of questions (object counting, relative distance, and relative direction) using structured templates. The algorithm takes as input an omni-cognitive map, a class count dictionary, and the expected max number of questions per type. We determine the expected maximum number of questions based on the similarity of the scenes to which the images correspond. For images obtained from DeepPanoContext [43], we set the expected number of questions to 10. For images obtained from ReplicaPano [10], we set the expected number of questions to 5.

To eliminate the duplication and ambiguity of the problem, we sample the objects in the scene without replacement and ensure the uniqueness of the positioning objects when generating non-negative sampling problems for relative distance and relative direction. In particular, if there are less than 5 object categories in the scene, we will not generate the above two types of problems. At the same time, we ensure that when generating relative direction problems, the three types of objects in the problem do not appear in the same cognitive map grid unit.

A.2. Negative Sampling for Hallucination Testing

Algorithm 3 outlines our negative sampling strategy for evaluating model robustness against hallucination. We first compute global statistics on object frequency (as shown in Figure 6) and co-occurrence patterns across the dataset. For each scene, we identify two types of distractor objects: ① frequently occurring objects that are absent from the current scene (\mathcal{M}_f) and ② objects that commonly co-occur with present objects but are missing from the current scene (\mathcal{M}_c). These negative distractor objects are incorporated into the object pool used for Omni-Cognitive map generation prompt and QA generation. Specifically, we set the number of \mathcal{M}_f and \mathcal{M}_c for each image to 5 when performing negative sampling.

Algorithm 2 QA Pair Generation

Require: Omni-cognitive map C , Class count dictionary \mathcal{D} , Questions per type N

- 1: **Define Question Templates:**
- 2: $T_{count} \leftarrow$ “How many *object*(s) are in this room?”
- 3: $T_{dist} \leftarrow$ “Measuring from the closest point of each object, which of these objects (*candidates*) is the closest to the *positioning* in this PANORAMA?”
- 4: $T_{dir} \leftarrow$ “If I am standing by the *positioning* and facing the closest *orienting*, is the closest *querying* to my front, back, left, right, front-left, front-right, back-left, or back-right in this PANORAMA?”
- 5: $\mathcal{O} \leftarrow$ all object classes in \mathcal{D}
- 6: $\mathcal{O}_{unique} \leftarrow o \in \mathcal{O} \mid \mathcal{D}[o] = 1$
- 7: $\mathcal{Q} \leftarrow \emptyset$
- 8: // Generate object counting questions
- 9: Sample $\min(N, |\mathcal{O}|)$ objects from \mathcal{O} without replacement
- 10: **for** each sampled object o **do**
- 11: $q \leftarrow T_{count}$ with *object* replaced by o
- 12: $a \leftarrow \mathcal{D}[o]$ //answer for o
- 13: Add to \mathcal{Q}
- 14: **end for**
- 15: // Generate relative distance questions
- 16: **for** $i = 1$ to N **do**
- 17: Sample positioning object p from \mathcal{O}_{unique}
- 18: Sample distinct candidates c_1, \dots, c_4 from $\mathcal{O} \setminus p$
- 19: $q \leftarrow T_{dist}$ with *positioning* replaced by p , *candidates* replaced by c_1, \dots, c_4
- 20: $a \leftarrow c_i$ find by Euclidean distance in grid coordinates
- 21: Add to \mathcal{Q}
- 22: **end for**
- 23: // Generate relative direction questions
- 24: **for** $i = 1$ to N **do**
- 25: Sample positioning object p from \mathcal{O}_{unique}
- 26: Sample distinct orienting object $o \in \mathcal{O} \setminus p$ and querying object $q \in \mathcal{O} \setminus p, o$
- 27: $q \leftarrow T_{dir}$ with *positioning* replaced by p , *orienting* replaced by o , and *querying* replaced by q
- 28: $a \leftarrow$ Direction Computed based on vector relationships in grid coordinates
- 29: Add to \mathcal{Q}
- 30: **end for**
- 31: **return** \mathcal{Q}

Algorithm 3 Negative Sampling for Hallucination Testing

Require: Omni-Cognitive map M , Parameters n, m

- 1: Compute global object frequency statistics \mathcal{F} and global co-occurrence statistics \mathcal{C}
- 2: Extract present objects \mathcal{P} from omni-cognitive map M
- 3: $\mathcal{M}_f \leftarrow$ Top n most frequent objects from \mathcal{F} that are not in \mathcal{P}
- 4: $\mathcal{M}_c \leftarrow \emptyset$ //Co-occurring but missing objects
- 5: $candidates \leftarrow \emptyset$
- 6: **for** each object $p \in \mathcal{P}$ **do**
- 7: **for** each object o co-occurring with p , sorted by co-occurrence frequency **do**
- 8: **if** $o \notin \mathcal{P}$ and $o \notin \mathcal{M}_f$ and $o \notin candidates$ **then**
- 9: $candidates \leftarrow candidates \cup o$
- 10: **if** $|candidates| = m$ **then**
- 11: **exit** loop
- 12: **end if**
- 13: **end if**
- 14: **end for**
- 15: **end for**
- 16: $\mathcal{M}_c \leftarrow$ Top m objects from $candidates$
- 17: $\mathcal{O}_{aug} \leftarrow \mathcal{P} \cup \mathcal{M}_f \cup \mathcal{M}_c$ //Augmented object set
- 18: Generate QA pairs using \mathcal{O}_{aug} with rules:
- 19: - For counting: Return 0 for objects in $\mathcal{M}_f \cup \mathcal{M}_c$
- 20: - For spatial questions: Return appropriate “not found” responses when needed
- 21: **return** QA pairs with both real and hallucinated objects

B. Dataset Detail

This section provides statistical insights into the composition and characteristics of our OSR-Bench dataset.

Our dataset contains a large number of question-answer pairs across different reasoning categories. For the standard dataset, we have a total of 74,326 questions distributed as follows:

- Relative distance questions: 26,240
- Object counting questions: 24,988
- Relative direction questions: 23,098

For the dataset with negative sampling, we have 78,986 questions distributed as:

- Object counting questions: 27,125
- Relative distance questions: 26,250
- Relative direction questions: 25,611

Figures 7-9 present the answer distributions for each question type, both with and without negative sampling.

C. Prompt Template

This section details the prompt templates used throughout our evaluation pipeline.

C.1. Omni-Cognitive Map Generate Prompt

The cognitive map generation prompt instructs models to analyze a panoramic image and construct a structured spatial representation of the scene. In actual application, in the case of non-negative sampling, we fill the {object_pool} with the object categories contained in the Ground truth. In the case of negative sampling, we add negative distractor objects to the {object_pool}.

Cognitive Map Prompt

[Task]

This PANORAMA captures an indoor scene. Your objective is to identify specific objects within the panorama, understand the spatial arrangement of the scene, and estimate the center point of each object, assuming the entire scene is represented by a 10x10 grid.

[Rule]

1. We provide the categories to care about in this scene: {object_pool}. Focus ONLY on these categories.
2. Estimate the center location of each instance within the provided categories, assuming the entire scene is represented by a 10x10 grid.
3. If a category contains multiple instances, include all of them.
4. Each object’s estimated location should accurately reflect its real position in the scene, preserving the relative spatial relationships among all objects.
5. Consider object distance from camera when positioning them. Closer objects should be placed near the center of the grid, while distant objects should be placed toward the edges.
6. If an object is partially visible or occluded, estimate its full position based on the visible parts.

[Output]

Present the estimated center locations for each object as a list within a dictionary. STRICTLY follow this JSON format And use INTEGERS within 10 to represent the coordinates: {"category name": [(x_1, y_1), ..., ...]}

C.2. Pre and Post Prompt

When we do not prompt the model to generate cognitive maps but ask it to answer questions directly, we add a pre-prompt before the question. The input for the models is formatted as follows: [Image Token] [Pre-prompt] [Question] [Post-prompt]

Figure 10 shows our complete reasoning process, which includes two evaluation modes: “Vanilla Mode” and “Think Mode”. We switch between the two modes by using different post-prompts.

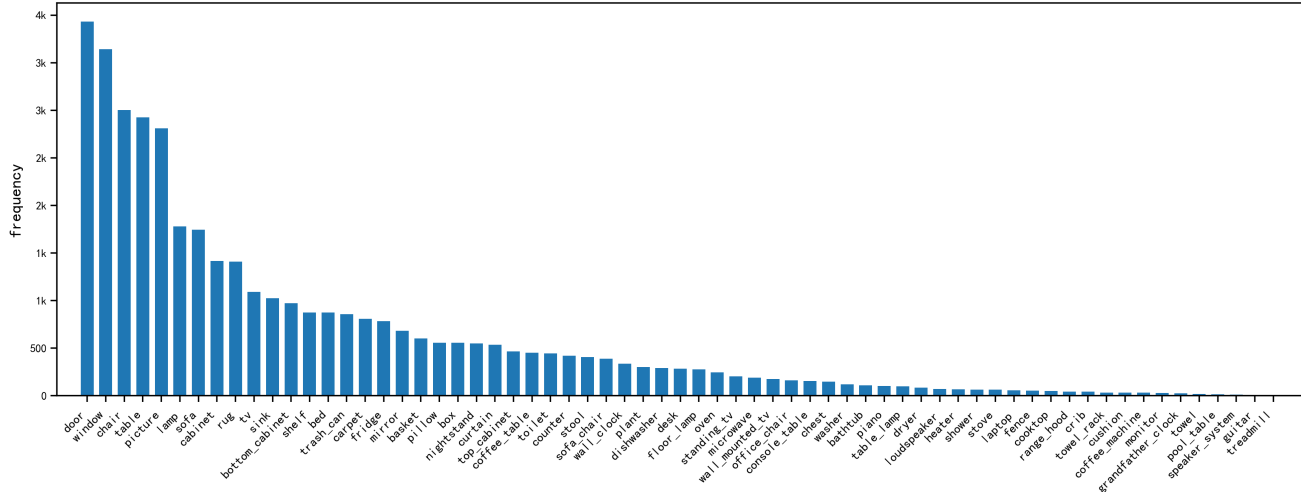


Figure 6. Object frequency of the entire dataset.

Pre-Prompt

According to the PANORAMA and the predicted objects' center locations, answer the following question:

text but has the correct number
Please respond in JSON format:
{
"score": 0 or 1,
"confidence": "high" or "middle" or "low"
}

C.3. LLM Evaluator Prompt

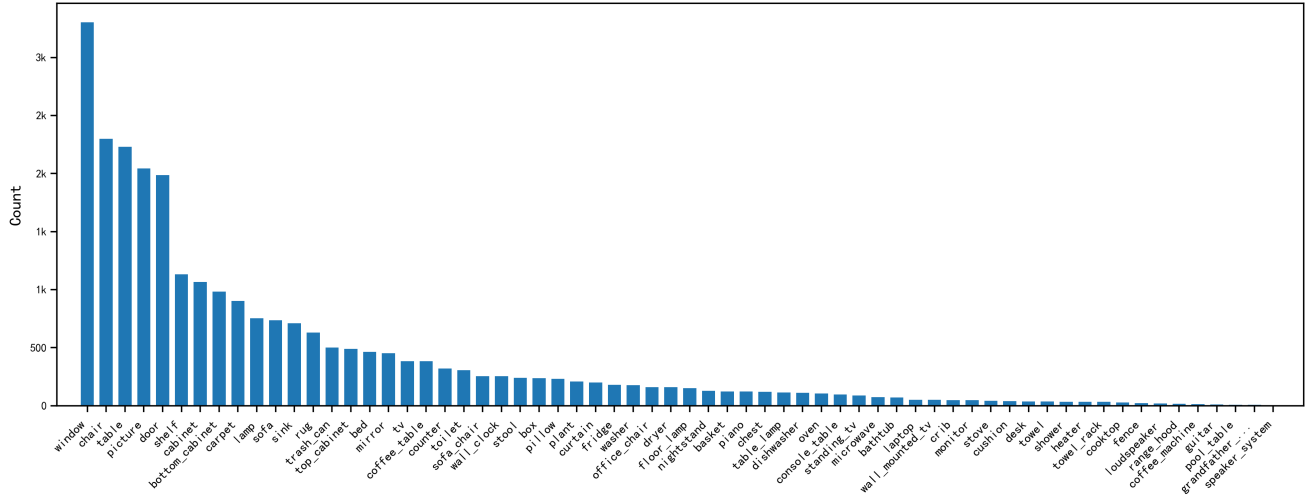
The LLM evaluator prompts enable automatic assessment of model responses, particularly for complex cases where rule-based evaluation may be insufficient. We provide three distinct evaluator prompts for the different question types. Each prompt includes specific evaluation rules and a structured JSON response format with both a binary score and confidence level. This approach ensures consistent assessment across diverse response formats from different models, especially for cases involving negative sampling where proper handling of non-existent objects is crucial. Specifically, we use DeepSeek-V3 as the evaluator, and replace {question}, {ground_truth}, {model_answer} in the template with the real case.

META_PROMPT_OBJECT_COUNTING

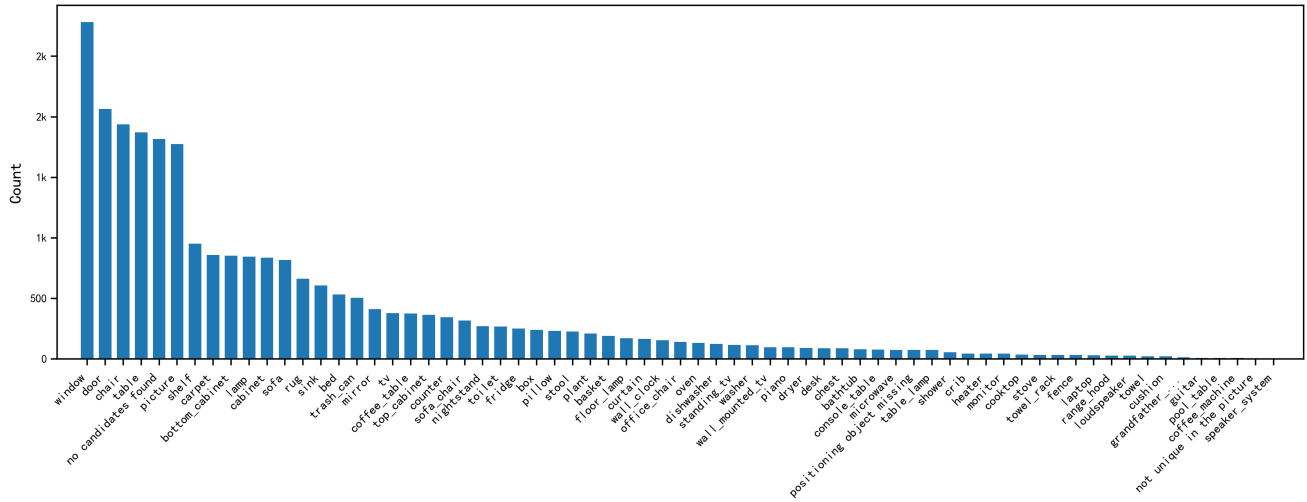
You are evaluating an object counting answer.
Question: {question}
Ground Truth: {ground_truth}
Model Answer: {model_answer}
Evaluation rules:
- The answer must be an integer (including 0)
- Correct if the model's integer exactly matches the ground truth
- Also correct if the model answer contains additional

META_PROMPT_RELATIVE_DISTANCE

You are evaluating a relative distance answer.
Question: {question}
Ground Truth: {ground_truth}
Model Answer: {model_answer}
Evaluation rules:
- Normal answers are object names
- Special case answers may be one of these error messages:
"The positioning object is not found in the picture."
"The positioning object is not unique in the picture."
"None of the candidates were found in the picture."
- When the ground truth is one of these special error messages, the model answer must match exactly
- When the ground truth is an object name, the model answer is correct if it contains the matching object name (case insensitive)
Please respond in JSON format:
{
"score": 0 or 1,
"confidence": "high" or "middle" or "low"
}



(a) QA pairs without negative sampling.



(b) QA pairs with negative sampling.

Figure 7. Answer distribution of Relative Distance question.

META_PROMPT_RELATIVE_DIRECTION

You are evaluating a relative direction answer.

Question: {question}

Ground Truth: {ground_truth}

Model Answer: {model_answer}

Evaluation rules:

- Normal answers are direction words: “Front”, “Back”, “Left”, “Right”, “Front-Left”, “Front-Right”, “Back-Left”, “Back-Right”
- Direction words are case insensitive, and hyphens are optional (e.g., “front left” equals “Front-Left”)
- Special case answers may be one of these error messages:

- “The positioning object is not found in the picture.”
- “The positioning object is not unique in the picture.”
- “The orienting object or querying object is not found in the picture.”
- When the ground truth is one of these special error messages, the model answer must match exactly
- Please respond in JSON format:

```
{
  "score": 0 or 1,
  "confidence": "high" or "middle" or "low"
}
```

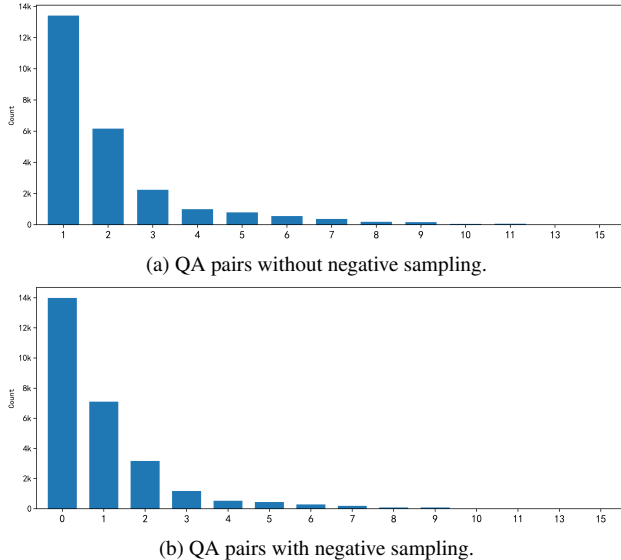


Figure 8. Answer distribution of Object Counting question.

D. Evaluation Details

To ensure reproducibility, we adopt a greedy decoding strategy for all models (*i.e.*, the temperature is set to 0, and both top-p and top-k are set to 1 for proprietary models, do_sample is set to False for open-source models).

A complete evaluation task (*i.e.*, evaluating a specific model with or without a omni-cognitive map, in vanilla or think mode, on a non-negative or negative sampling dataset) requires approximately 4-5 days. For hardware requirements, models below 13B parameters use 1 NVIDIA A40 48G GPU, models between 13B and 30B use 3 NVIDIA A40 48G GPUs, while models exceeding 70B parameters require 4 NVIDIA A800 80G GPUs.

E. Additional Results on Newer MLLMs

To make the experiments more complete, we additionally evaluate three newer post-submission MLLMs—GPT-5.2, Gemini-3-flash, and Qwen3-VL-A22B—under the same protocol as in the main paper. These results are intended as a supplementary check rather than a new benchmark split, and they follow the same metrics and prompts as Tables 2 and 3.

Table 6. Omni-cognitive map generation results for newer MLLMs under the same setup as Table 2.

Model	Avg dist.↓	Precision	Recall	F1	Succ. Rate(%)
GPT-5.2	3.544	0.276	0.224	0.243	99.73
Gemini-3-flash	14.92	0.002	0.002	0.002	1.67
Qwen3-VL-A22B	4.11	0.291	0.221	0.247	94.83

Table 6 shows that newer models still struggle to build accurate omni-cognitive maps. GPT-5.2 and Qwen3-VL-

A22B produce well-formatted maps reliably, but their spatial precision remains limited. Gemini-3-flash frequently outputs object-aligned 2D boxes rather than a coherent allocentric map, which causes a very low success rate under our format and matching criteria.

Table 7. Question-answering accuracy of newer MLLMs without negative sampling, using the same setting as Table 3.

Model	object count		relative distance		relative direction	
	w/ cogmap	w/o cogmap	w/ cogmap	w/o cogmap	w/ cogmap	w/o cogmap
GPT-5.2	0.798	0.750	0.450	0.522	0.252	0.307
Gemini-3-flash	0.827	0.769	0.500	0.540	0.168	0.142
Qwen3-VL-A22B	0.790	0.641	0.416	0.420	0.291	0.252

As shown in Table 7, these newer models modestly improve absolute scores, but the overall pattern remains unchanged. Object counting is the easiest task, relative direction remains the hardest, and gains from stronger models do not remove the large gap to robust omnidirectional spatial reasoning. Taken together with the ground-truth cognitive map results in Section F, these findings further support our main conclusion that spatial representation extraction remains the core bottleneck.

F. More Cases

F.1. Effect of Ground-Truth Cognitive Maps

In the main paper (Table 3), the cognitive maps are always generated on-the-fly by the evaluation models themselves, rather than being provided as ground truth. This design choice allows us to assess the end-to-end spatial reasoning ability of MLLMs, including both spatial representation extraction and downstream reasoning.

To further isolate the impact of the spatial representation, we additionally evaluate two strong MLLMs—Gemini-1.5-Pro and Qwen2.5-VL-72B—under three settings:

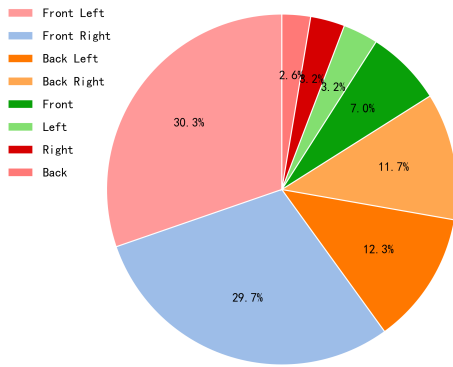
- Generated Cogmap:** cognitive maps generated by the model itself (same as Table 3).
- GT Cogmap (w/o CoT):** ground-truth cognitive maps are provided, and the model answers in vanilla mode.
- GT Cogmap (w/ CoT):** ground-truth cognitive maps are provided, and the model answers in think mode with chain-of-thought reasoning.

Table 8 reports the resulting accuracies for each task.

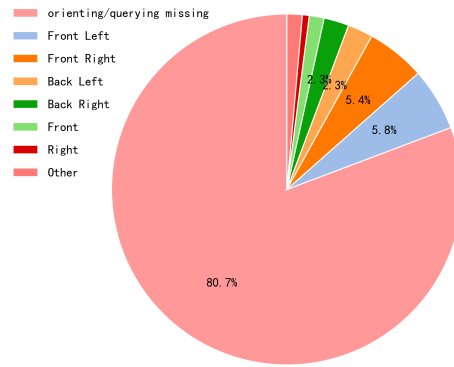
Table 8. Impact of using ground-truth (GT) cognitive maps on spatial reasoning performance.

Model	Task	Generated	GT (w/o CoT)	GT (w/ CoT)
Gemini-1.5-Pro	Object Count	0.744	0.914 (+0.170)	0.957 (+0.213)
	Relative Distance	0.417	0.514 (+0.097)	0.623 (+0.206)
	Relative Direction	0.074	0.168 (+0.094)	0.286 (+0.212)
Qwen2.5-VL-72B	Object Count	0.558	0.688 (+0.130)	0.774 (+0.216)
	Relative Distance	0.299	0.409 (+0.110)	0.417 (+0.118)
	Relative Direction	0.156	0.271 (+0.115)	0.332 (+0.176)

We observe substantial improvements across all tasks when ground-truth cognitive maps are supplied. Object



(a) QA pairs without negative sampling.



(b) QA pairs with negative sampling.

Figure 9. Answer distribution of Relative Direction question.

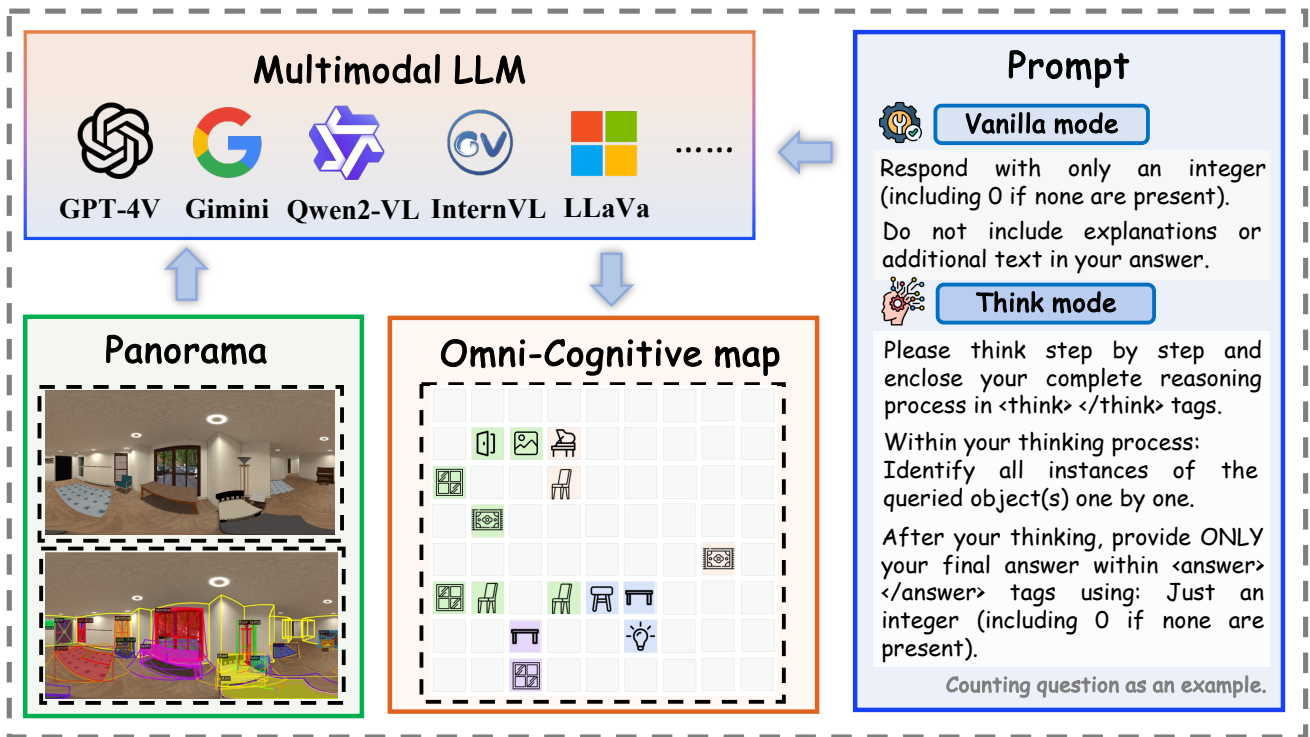


Figure 10. Inference pipeline.

counting benefits the most in absolute terms—Gemini-1.5-Pro reaches near-perfect accuracy (0.957) in think mode. Relative direction, which exhibits the lowest performance in the fully end-to-end setting, shows the largest relative gain from GT cognitive maps. In all cases, think mode consistently outperforms vanilla mode, extending the findings of Table 8 in the main paper to the GT-cogmap regime.

These results indicate that accurate spatial representation extraction is currently the primary bottleneck for om-

nidirectional spatial reasoning: once provided with precise cognitive maps, MLLMs can reason much more reliably. The qualitative examples of think mode using ground-truth cognitive maps are shown in Figure 13.

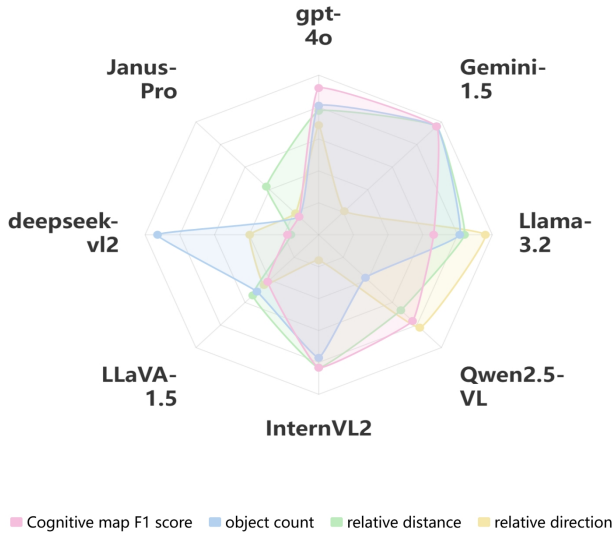


Figure 11. *OSR*-benchmark Overview

F.2. Human Evaluation

To complement the automatic evaluation, we conduct a human study to quantify human performance on OSR-Bench. We randomly sample 100 spatial reasoning problems corresponding to different panoramic images: 30 object counting questions, 30 relative distance questions, and 40 relative direction questions. Three human evaluators independently answer the questions given the panoramic images, and we compute accuracy using the same criteria as for MLLMs.

To protect participant privacy, we report only anonymized, aggregated results and do not disclose any personally identifying information.

Table 9 summarizes the results and compares them with the best-performing MLLM in our benchmark (Gemini-1.5-Pro) on the same subset.

Humans substantially outperform MLLMs on all three task types, confirming that there remains a considerable gap between current MLLMs and human-level omnidirectional spatial reasoning. At the same time, human accuracy also decreases as task difficulty increases (from counting to distance to direction), which validates that OSR-Bench poses intrinsically challenging spatial reasoning problems even for human annotators.

Table 9. Human evaluation on a subset of OSR-Bench. We report accuracy for three human evaluators and the best-performing MLLM (Gemini-1.5-Pro) on the same 100 problems.

Task Type	Human 1	Human 2	Human 3	Gemini-1.5-Pro
Object Count	0.926	0.917	0.892	0.732
Relative Distance	0.800	0.733	0.867	0.437
Relative Direction	0.775	0.650	0.725	0.273

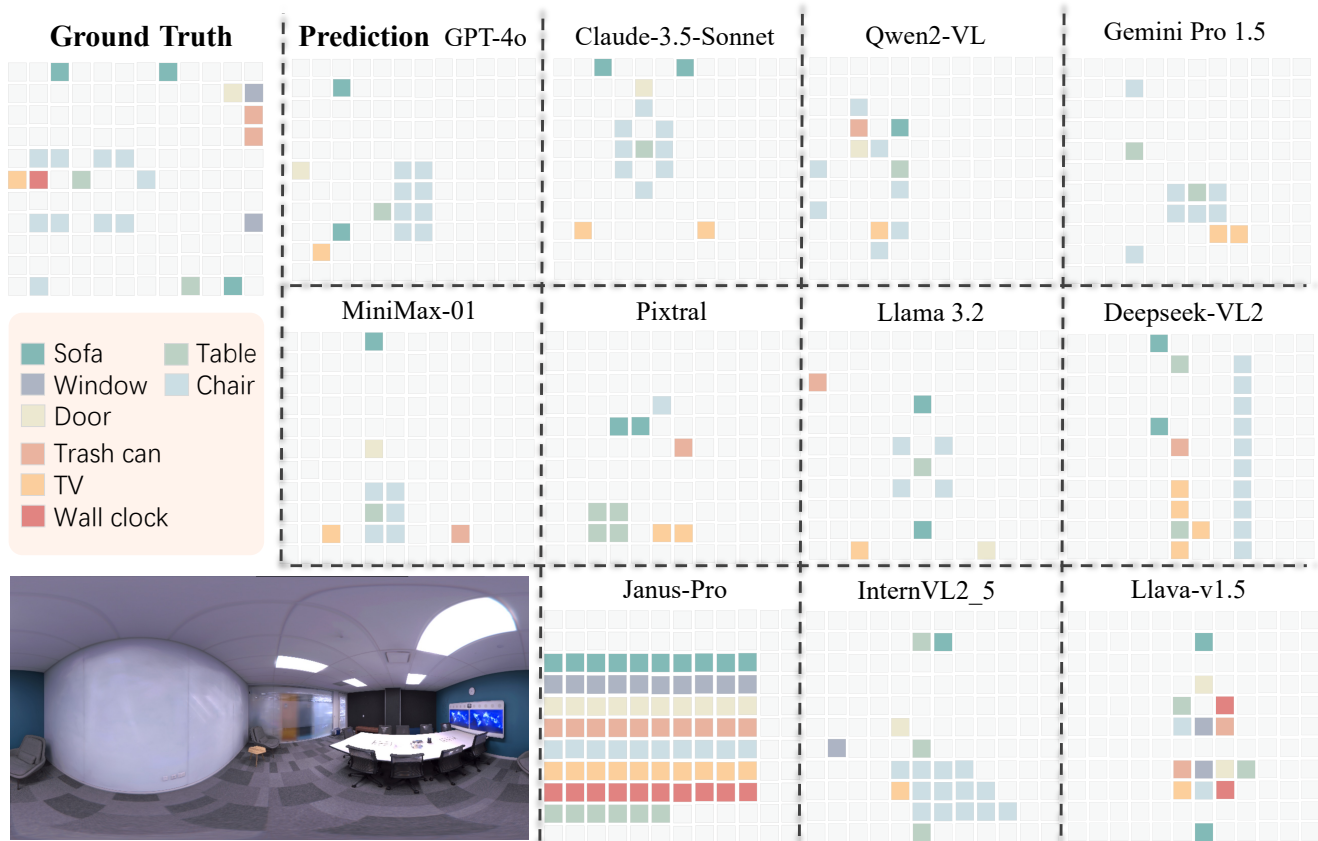


Figure 12. Visualization of Omni-Cognitive map generation

Relative distance

Question:
Measuring from the closest point of each object, which of these objects (**cabinet**, **tv**, **rug**, **chair**) is the **closest to the mirror** in this PANORAMA?

Answer:
<think>
The objects to consider are the cabinet, TV, rug, and chair. The reference object is the mirror.
1. **Mirror:** Located on the right side of the panorama.
2. **Cabinet:** Located right next to the mirror, on its right/below in the image. This is likely the closest object.
3. **TV:** There appears to be no TV in this panorama, so it cannot be the closest.
4. **Rug:** There are two rugs visible, one below the dining table and one on the other side of the room. Both rugs are further than the cabinet from the mirror.
5. **Chair:** There are multiple chairs visible, some around the dining table and some near the "other table" in the image. The chairs around the dining table are further than the mirror.
Since the cabinet is directly adjacent to the mirror, it is the closest.
</think>
<answer>**cabinet**</answer>

Object counting

Question:
How many **door(s)** are in this room?

Answer:
<think>
The provided categories are {mirror, cabinet, window, chair, table, rug, door, tv}. We are looking for instances of "door". Examining the panorama, there's one apparent door on the left side, leading to a dark hallway. This is at coordinate [1, 4] approximately. It is partially occluded on the left but appears to be a single full door structure. No other objects in the scene resemble a door according to the usual definition.
</think>
<answer>**1**</answer>

Relative direction

Question:
If I am standing by the **door** and facing **the closest mirror**, is **the closest rug** to my front, back, left, right, front-left, front-right, back-left, or back-right in this PANORAMA?

Answer:
<think>
1. **Positioning Object:** I am standing by the door. The door is located at [1, 4].
2. **Facing Direction:** I am facing the closest mirror. The mirror is at [9, 5]. This means I am facing right.
3. **Target Objects:** The rugs are located at [1, 2] and [9, 8].
4. **Closest Rug:** The rug at [1, 2] is closer to the door [1, 4] than the rug at [9, 8].
5. **Relative Direction:** The closer rug [1, 2] is to the left of the door [1, 4], since its y-coordinate is smaller. Since I'm facing right, anything left with respect to my starting position at the door would be to my back-Left.
</think>
<answer>**Back-Left**</answer>

Figure 13. Good case in think mode using ground-truth cognitive maps