

360° Image Perception with MLLMs: A Comprehensive Benchmark and a Training-Free Method (Supplementary Material)

Huyen. T. T. Tran¹, Van-Quang Nguyen¹ , Farros Alferro¹, Kang-Jun Liu¹
, and Takayuki Okatani^{1,2}

¹ Department of System Information Sciences,
Graduate School of Information Sciences, Tohoku University, Japan

² RIKEN Center for AIP

{tran, quang, farrosalferro, kjliu, okatani}@vision.is.tohoku.ac.jp

In this supplementary material, we present additional information supporting the main manuscript:

- Section 1 further describes the 360Bench benchmark, including viewer-centric direction visualization, the annotation protocol, and benchmark statistics. It also discusses the unique challenges of 360° spatial reasoning and defines the scope and hierarchical task taxonomy of 360Bench.
- Section 2 elaborates on the Free360 framework and its implementation.
- Section 3 details the human evaluation study, experimental settings, and full evaluation results for 27 models.
- Section 4 further analyzes the performance of Free360. It also discusses the trade-off between training-free and fine-tuned methods, the mitigation of error propagation, potential strategies to improve efficiency, and qualitative results.

1 Additional Details on 360Bench

1.1 Viewer-centric Direction Visualization

360° images pose a challenge for models to perform spatial reasoning from the viewer’s perspective, as illustrated in Fig. 1(a). To better represent the spatial layout of ERP images, we utilize the CMP format, which divides a 360° scene into six cube faces. By assigning a unique color to each face and converting the CMP image back into ERP, we produce the visualization of viewer-centric directions shown in Fig. 1(b). This visualization clarifies how six principal directions relative to the viewer (i.e., top, bottom, left, right, front, and behind) are mapped onto ERP images. For example, the front view is centered, while the behind view is split across the left and right edges of the image.

1.2 Annotation Protocol

We developed a 360° annotation user interface (UI) to facilitate efficient 360° image exploration, bounding box labeling, and question-answer annotation, as shown in Fig. 2.

Overview and Exploration. Annotators could browse the benchmark, select images to annotate, and review previously saved annotations to avoid duplication. To support 360° spatial reasoning, black lines were overlaid in the images to indicate spatial layouts corresponding to the six principal viewer-centric directions, as illustrated in Fig. 1(a). Annotators could view the 360° images in Virtual Reality (VR) mode by clicking the *Open in 360 Viewer* button, adjust the viewpoint by dragging, and zoom in or out by scrolling. The six screenshots in the top-right of Fig. 2 show six viewpoints in the VR mode. Annotators were encouraged to explore each image in both non-VR mode (ERP) and VR mode to better locate distorted or hard objects.

Question-Answer Annotation. Annotators selected (sub)task categories based on the objects’ characteristics and subsequently formulated corresponding question-answer pairs. AI-assistant functions (Gemini-2.0-flash [19]) were used to provide grammar corrections and suggest incorrect answer options to improve annotation efficiency. These suggestions typically exhibited strong semantic relevance to the questions but weak correspondence to the visual content. Annotators were therefore required to review the suggestions carefully and rectify them to create stronger hard-negative distractors.

Bounding Box Labeling. Bounding boxes were drawn on ERP images. Annotators could zoom in or out to accurately label objects, as shown in the bottom-right of Fig. 2. The labeling process was performed in two steps. First, annotators defined context-aware labels that describe both the object category and its relevant context. Next, they drew the corresponding bounding boxes on the image. For example, in Fig. 2, given the question “How many people are sitting in front of the entrance to the Hôtel de Ville?”, the annotator specified the label “person sitting in front of the entrance to the Hôtel de Ville” and then outlined the associated bounding boxes on the canvas below.

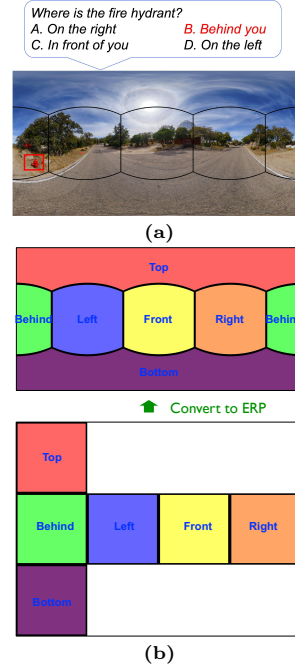


Fig. 1: (a) SR-OV example. Image source: [17]. (b) Viewer-centric direction visualization.

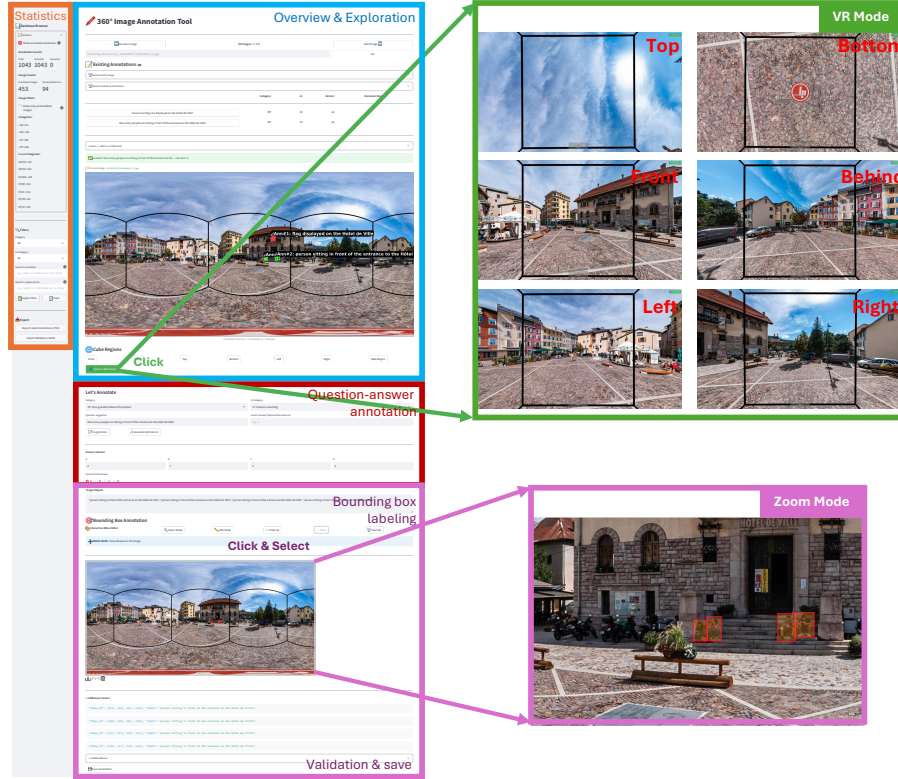


Fig. 2: User interface for 360° image annotation, supporting VR exploration, question-answer annotation, and bounding-box labeling. Original 360° image source: [5].

Validation and Save. Once the bounding boxes were annotated, they were displayed in JSON format for validation. Although not shown in the figure, simple rule-based checks were applied to ensure format consistency (e.g., questions end with a question mark), and warning messages were displayed in the UI when violations occurred. After resolving any issues, annotators could then save the completed annotations.

1.3 Statistics on 360Bench

Word Frequency Analysis. Figure 3 presents five word clouds from our 360Bench, offering a concise overview of its task coverage. Figures 3(a)–(e) correspond to the full benchmark and four tasks: fine-grained perception (FP), projection-distorted perception (PP), spatial reasoning (SR), and direction-giving (DG). Word size reflects term frequency.

Directional terms such as left, right, behind, and front dominate in Figs. 3(a) and (d), underscoring the strong focus on spatial reasoning—a central challenge

images where the field of view is limited. In contrast, 360° images can cover a complete view of the surrounding environment within a single frame, allowing for comprehensive spatial reasoning across the scene. Furthermore, the geometric distortions and object fragmentation caused by sphere-to-plane projections make spatial reasoning in 360° images significantly more challenging. Stretched and split objects and bent lines fundamentally disrupt standard spatial priors that MLLMs have acquired from conventional images.

1.5 Viewpoint Reasoning Scope

In 360Bench, both the object-to-viewer reasoning (SR-OV) and direction-giving (DG) tasks initialize the viewpoint at the center of the ERP image. However, while SR-OV assumes a fixed viewpoint throughout, the DG task requires the model to implicitly update this viewpoint across sequential navigation actions (as illustrated in Fig. 10(c) of the supplementary material). Arbitrary-viewpoint reasoning, where the viewpoint can be initialized at any location, falls outside the scope of this work and will be explored in future research.

1.6 Task Taxonomy of 360Bench

To systematically evaluate MLLM capabilities in perceiving 360° images, 360Bench introduces a taxonomy of seven subtasks grouped into four progressive reasoning levels. This hierarchical structure advances from foundational visual grounding to complex action planning, ensuring broad coverage of real-world challenges.

The first level, foundational perception, comprising the FP-IR and FP-IC subtasks, assesses the ability to accurately identify and localize entities within a scene. Building upon this capability, the second level, distortion robustness, including the PP-IR and PP-IC subtasks, evaluates whether models can maintain reliable perception under the severe geometric distortions and object fragmentation introduced by sphere-to-plan projections. The third level, comprehensive spatial reasoning, examines both allocentric reasoning over object-to-object relationships (SR-Os) and egocentric reasoning from the viewer’s perspective (SR-OV). Finally, the fourth level, action planning, represented by the DG task, requires models to integrate perceptual and spatial information to perform multi-step route planning and navigation.

Together, these seven subtasks form a coherent evaluation framework that progresses from entity-level perception (FP and PP), to relational spatial understanding (SR), and ultimately to goal-oriented planning (DG). This taxonomy enables 360Bench to comprehensively assess the diverse capabilities required for real-world panoramic scene understanding.

2 Additional Details on Free360 Framework

2.1 Prompt Templates

We provide the prompt templates used in Free360 in Tables 2–4. Each template corresponds to one step of Free360, i.e., entity identification, attribute extrac-

tion, inter-entity relation detection, and scene graph-based question answering. A template takes either a 360° image-question pair or intermediate outputs from the previous step as input. In the question answering step, the in-context examples are (sub)task-agnostic, allowing the model to handle diverse queries. Each template specifies a system prompt, step-by-step reasoning instructions, and a clear output format to keep the model aligned with its role at that stage. If the model fails to comply with the required format, its output is treated as an empty string.

2.2 Spherical Rotation

To detect the inter-entity relations (Step 3 in Free360), we apply spherical rotations on the original CMP image to center each entity pair. For this purpose, we employ a spherical coordinate system that follows the right-hand rule [9], as illustrated in Fig. 5(a). The CMP image has resolution $W \times H$ pixels, with the origin (0,0) of the 2D coordinate system at the top-left corner. Following [9], we also define 2D local (u, v) coordinate systems on the cube faces, as shown in Fig. 5(b). The width and height of each cube face are $W_f = W/4$ and $H_f = H/3$, respectively.

To center the entity pair (n_i^e, n_j^e) , the spherical rotation is determined by ϕ' (rotation about the Y-axis) and θ' (rotation about the Z-axis), which are derived from the bounding boxes of the two entities. Let $\mathcal{C}$ denote the set of all eight corners of these bounding boxes:

$$\mathcal{C} = \{(x_k^e, y_k^e) \mid k = 1, 2, \dots, 8\}.$$

We define the center $\mathbf{c}^*$ of the entity pair as the midpoint of the minimal spherical bounding box (MSBB) that encloses both entities. To determine the MSBB, we first map the 2D coordinates of each bounding-box corner (x_k^e, y_k^e) to its corresponding cubemap face index s_k and to the local coordinates (u_k, v_k) on that face, using an approximate mapping for simplicity:

$$s_k = \begin{cases} 2, & \text{if } y_k^e < \frac{H}{3} \\ 3, & \text{if } y_k^e \geq \frac{2H}{3} \\ 1, & \text{if } x_k^e < \frac{W}{4} \\ 4, & \text{if } x_k^e \in [\frac{W}{4}, \frac{W}{2}) \\ 0, & \text{if } x_k^e \in [\frac{W}{2}, \frac{3W}{4}) \\ 5, & \text{otherwise.} \end{cases} \quad (1)$$

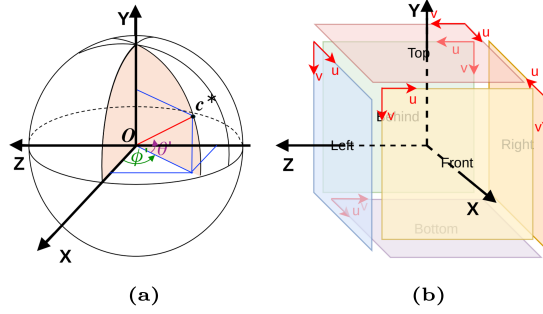


Fig. 5: (a) Spherical coordinate system. (b) 2D coordinate system on cube faces.

$$(u_k, v_k) = \begin{cases} (H_f - y_k^e, x_k^e), & \text{if } s_k = 2 \\ (y_k^e - 2H_f, \max(0, W_f - x_k^e)), & \text{if } s_k = 3 \\ (x_k^e, y_k^e - H_f), & \text{if } s_k = 1 \\ (x_k^e - W_f, y_k^e - H_f), & \text{if } s_k = 4 \\ (x_k^e - 2W_f, y_k^e - H_f), & \text{if } s_k = 0 \\ (x_k^e - 3W_f, y_k^e - H_f), & \text{otherwise.} \end{cases}$$

Next, the local coordinates (u_k, v_k) are normalized to the range $[-1, 1]$ and then mapped to the corresponding 3D coordinates on the sphere (X_k, Y_k, Z_k) as follows:

$$(\hat{u}_k, \hat{v}_k) = \left(\frac{2(u_k + 0.5)}{W_f} - 1, \frac{2(v_k + 0.5)}{H_f} - 1 \right), \quad (2)$$

$$(X_k, Y_k, Z_k) = \begin{cases} (1, -\hat{v}_k, -\hat{u}_k), & \text{if } s_k = 0, \\ (-1, -\hat{v}_k, \hat{u}_k), & \text{if } s_k = 1, \\ (\hat{u}_k, 1, \hat{v}_k), & \text{if } s_k = 2, \\ (\hat{u}_k, -1, -\hat{v}_k), & \text{if } s_k = 3, \\ (\hat{u}_k, -\hat{v}_k, 1), & \text{if } s_k = 4, \\ (-\hat{u}_k, -\hat{v}_k, -1), & \text{otherwise.} \end{cases} \quad (3)$$

The corresponding longitude ϕ_k and latitude θ_k of each bounding-box corner are calculated by

$$\phi_k = \arctan 2(-Z_k, X_k) \bmod 2\pi, \quad (4)$$

$$\theta_k = \arcsin \left(\frac{Y_k}{\sqrt{X_k^2 + Y_k^2 + Z_k^2}} \right). \quad (5)$$

The latitude bounds of the MSBB, denoted by θ_l (lower bound) and θ_u (upper bound), are given by the minimum and maximum latitudes of the corners:

$$\theta_l = \min_{k \in \{1, \dots, 8\}} \theta_k, \quad \theta_u = \max_{k \in \{1, \dots, 8\}} \theta_k.$$

Computing the longitude bounds requires handling wrap-around at 0 and 2π . To achieve this, we sort the longitudes of the corners and compute the gaps between consecutive values, including the wrap-around gap. The largest gap is identified, and the minimal longitudinal interval, from ϕ_l to ϕ_u , is defined as the complement of this gap. The longitude and latitude of the center point $\mathbf{c}^*$ are then computed as follows:

$$\phi_{\mathbf{c}^*} = \begin{cases} (\phi_l + \phi_u)/2, & \text{if } \phi_l \leq \phi_u \\ ((\phi_l + \phi_u + 2\pi)/2) \bmod 2\pi, & \text{otherwise.} \end{cases} \quad (6)$$

$$\theta_{\mathbf{c}^*} = (\theta_l + \theta_u)/2. \quad (7)$$

The rotation angles (ϕ', θ') are set to the longitude and latitude of $\mathbf{c}^*$, respectively. The resulting rotation matrix $\mathbf{R}_c$ is given by

$$\mathbf{R}_c = \begin{bmatrix} \cos \phi' \cos \theta' & -\cos \phi' \sin \theta' \sin \phi' \\ \sin \theta' & \cos \theta' & 0 \\ -\sin \phi' \cos \theta' & \sin \phi' \sin \theta' & \cos \phi' \end{bmatrix}.$$

2.3 Entity-View Mapping Function

To determine the entity-view relations in Step 4 of Free360, we define an approximate mapping function $f: \mathbb{R}^2 \rightarrow \mathcal{N}^v$ that maps the 2D pixel coordinates (x, y) in the CMP image to the corresponding view node n_q^v . For an entity node with a 2D bounding box $(x_{i,1}^e, y_{i,1}^e, x_{i,2}^e, y_{i,2}^e)$, the center coordinates are computed as

$$\bar{x}_i = \frac{(x_{i,1}^e + x_{i,2}^e)}{2}, \quad (8)$$

$$\bar{y}_i = \frac{(y_{i,1}^e + y_{i,2}^e)}{2}. \quad (9)$$

The mapping function is defined as follows:

$$f(x, y) = \begin{cases} \text{top view,} & \text{if } y < \frac{H}{3} \\ \text{bottom view,} & \text{if } y \geq \frac{2H}{3} \\ \text{behind view,} & \text{if } x < \frac{W}{4} \\ \text{left view,} & \text{if } x \in [\frac{W}{4}, \frac{W}{2}) \\ \text{front view,} & \text{if } x \in [\frac{W}{2}, \frac{3W}{4}) \\ \text{right view,} & \text{otherwise.} \end{cases} \quad (10)$$

3 Additional Details on Experiments

3.1 Human Evaluation Study

To quantify the human performance, we conducted a subjective test with five participants, each completing a set of 70 questions (10 per subtask) randomly sampled from 360Bench. We built a web-based user interface based on A-Frame [1], enabling the participants to explore 360° scenes using a mouse. Each session comprised three sequential phases: introduction, training, and testing. Each participant completed the session in approximately 20–50 minutes.

Introduction Phase. The participants were provided with comprehensive information regarding the test, its subtasks, and instructions, as shown in Fig. 6(a). Upon understanding the instructions, they proceeded to the training phase via the *I'm Ready to Begin* button.

Training Phase. In this phase, each participant answered seven training questions (one per subtask) to get familiar with the 360° images, question formats, and the user interface. These answers were not included in the final scoring.

Fig. 6(b, bottom-left) shows a screenshot during the training phase. The top panel presents the participants’ progress, reflecting the total questions completed and the number completed within each subtask. While viewing the 360° images, the participants could freely adjust their viewpoints by dragging, zoom in or out by scrolling, and toggle the visibility of the question panel with the right-click. For each question, the corresponding subtask was also displayed within the question panel to help participants clearly understand the type of reasoning required. To further support spatial reasoning, a red arrow at the bottom of the screen shows the current viewing direction in real time. Once an answer was recorded, the next image-question pair was presented.

To alleviate motion sickness during prolonged 360° viewing, the participants could pause the test by clicking the *Pause* button to take short breaks. While the test was paused, the image was darkened, the question panel was hidden, and all navigation controls were disabled.

Testing Phase. After completing the training phase, the participants advanced to the testing phase, as illustrated in Fig. 6(b, bottom-right). In this phase, each participant answered 70 questions (10 per subtask), which were randomly sampled from 360Bench and distinct from the training questions. They followed the same procedure for answering questions as in the training phase. However, unlike in the training phase, the subtasks were not disclosed to prevent participants from gaining potential hints. Once all 70 questions were completed, the results were presented and saved for post-experimental analysis.

Following the completion of the test, we collected the results from the five participants to compute the average score and the mean response time per question, reporting these values as indicators of human effectiveness and efficiency.

3.2 Experimental Settings

In this experiment, we evaluate 27 models, including both MLLMs and enhancement variants. Specifically, we assess 20 representative proprietary and open-source MLLMs: GPT-4o [18], Gemini Flash 2.5 [22], Gemini Pro 2.5 [22], Qwen2.5-VL-3B [2], Qwen2.5-VL-7B [2], Qwen2.5-VL-32B [2], DeepSeek-VL2-Small [31], LLaVA-v1.5-7B [14], LLaVA-NeXT-7B [15], LLaVA-OV-0.5B [13], LLaVA-OV-7B [13], LLaVA-CoT [32], Idefics-9B [11], Idefics2-8B [12], Idefics3-8B [10], GLM-4V-9B [23, 25], Llama-3.2-Vision-11B [16], Emu3 [28], Molmo-7B-D [3], and Molmo-7B-O [3]. In addition, we evaluate seven enhancement methods: SEAL [30], DC² [26], ZoomEye [21], VisCoT-7B-336 [20], DeepEyes [35], Omni-CoT [33], and Free360.

Experiments were conducted on a computing node with 8 cores of an Intel Xeon Platinum 8558 processor, 160 GB of RAM, and an NVIDIA H200 GPU. To ensure consistency and reproducibility, all models, except proprietary MLLMs, were configured to generate outputs deterministically using greedy decoding. We

evaluated the MLLMs using VLMEvalKit [4], an open-source toolkit for model evaluation.

3.3 Full Evaluation Results

Table 1 compares the performance and inference time of humans and 27 models evaluated on 360Bench.

Human evaluation reveals strong performance on recognition (FP-IR), spatial reasoning (SR-Os), and direction-giving (DG) tasks, achieving average accuracies of at least 84% with low inter-participant variability (i.e., standard deviations $\leq 8\%$). In contrast, counting tasks (FP-IC and PP-IC) are substantially more challenging, resulting in lower accuracies (at most 82%) and notably higher standard deviations (11.7% and 17.2%, respectively).

Of all models, Gemini Pro 2.5 achieves the highest accuracy (46.5%) but remains far below human performance (86.3%). It should be noted that, owing to their black-box nature, comparisons of Gemini Pro 2.5 or other proprietary models with open-source models may be less informative. Among open-source models, Qwen2.5-VL performs best, achieving 38.1% at 7B and 39.7% at 32B. For enhancement methods, ZoomEye attains the highest performance on FP-IR (61.5%), while our Free360 achieves the highest accuracy across all other tasks and the best overall performance (45.3%).

4 Further Analysis of Free360’s Performance

4.1 Training-Free vs. Fine-Tuning

We fine-tune Qwen2.5-VL-7B with LoRA [6] under two settings: (S1) 10 and (S2) 50 training samples per subtask $\times$ 7 subtasks, using the rest for testing and repeating the process over 5 random seeds. These fine-tuned models achieve fast inference times of 0.4 seconds and 0.5 seconds but only reach 36.6% and 41.3% accuracy, respectively. Meanwhile, our training-free Free360 framework achieves 45.4% accuracy on both test sets with a 22.4s latency. This highlights a trade-off: fine-tuning offers faster inference but requires task-specific data and performs worse in low-data regimes, whereas Free360 achieves higher accuracy without additional training.

4.2 Mitigating Error Propagation in Free360

Because Free360 operates as a sequential four-step pipeline, incorrect entity detection in Step 1 (Entity Identification) can cause subsequent steps to fail, yielding a scene graph that is insufficient for answering the question. To mitigate this error propagation, the original image serves as a fallback input to support the final answer generation when the generated scene graph lacks the necessary details.

To investigate the effectiveness of the fallback mechanism, we evaluate two extreme settings. First, randomizing bounding boxes yields 36.3% accuracy, which

remains close to the base MLLM (38.1%). Second, using ground-truth boxes achieves 53.7% accuracy. These findings indicate a strong reliance on Step 1, with the fallback mechanism effectively mitigating downstream degradation.

To estimate the fraction of failures due to Step 1, we define a failure as Step-1-related if any ground-truth entity lacks a matched prediction or has $\text{IoU} < 0.5$. This criterion accounts for 72.2% of failures. For Step 3, we analyze failed SR-Os samples, as this task requires inter-entity relation reasoning. A failure is attributed to Step 3 if Step 1 succeeds, accounting for 31.8% of SR-Os failures.

4.3 Effectiveness of Spherical Rotation in Step 3

Entity-centered rotation is a core 360°-specific operation in Free360. Because it primarily targets inter-entity relation detection, its modest overall accuracy contribution (+0.7%) underrepresents its local impact. Removing it yields a 2.8% accuracy drop on SR-Os and a 1.4% drop on FP. The FP gain is secondary and mainly occurs when small targets are spatially anchored by surrounding landmarks, e.g., flags on/around a stadium (see Fig. 8(b)). When distributed across all seven subtasks, these localized gains average to $(2.8 + 2 \times 1.4) / 7 \approx 0.8\%$, which is consistent with the observed overall gain.

4.4 Comparison with Omni-CoT [33].

Unlike Omni-CoT, which relies on a sequential refinement pipeline using view descriptions and ERP-based object crops, Free360 constructs a question-relevant scene graph. This graph explicitly integrates entity nodes, cropped attributes, entity-view, inter-view, and inter-entity relations into a unified reasoning scaffold. This structural difference remains even without rotation: Free360 w/o Rotate achieves 44.7%, still +5.2% over Omni-CoT (39.5%). It is noted that Omni-CoT was evaluated using its full pipeline, including response refinement, under the same base model and evaluation protocol.

The superior performance of Free360 stems from its structured graph aggregation and explicit 360° relation modeling. Specifically, CMP-based entity detection enhances robustness against distortion, while entity-view links facilitate viewer-centric reasoning. These advantages are reflected in the gains over Omni-CoT on counting and spatial tasks, e.g., FP-IC +5.7%, PP-IC +3.6%, PP Avg. +4.4%, SR-Os +5.3%, SR-OV +12.3%, and DG +6.4%.

At the component level, Free360 capitalizes on CMP-ERP complementarity by detecting objects directly on the full CMP image (Step 1) and capturing spatial relations among entities on the ERP image (Step 3), whereas Omni-CoT relies on a combination of six perspective views and an ERP representation. Furthermore, while Omni-CoT links view descriptions to orientations and relies on implicit crop-orientation alignment, Free360 explicitly associates each detected entity with its orientation.

Overall, Free360 is a unified, scene-graph-based reasoning framework that is different from Omni-CoT’s sequential refinement strategy. Ultimately, the two

approaches diverge significantly in their representation, reasoning mechanisms, and handling of 360° spatial challenges.

4.5 Inference Time and Potential Strategies for Improvement

Inference Time. The superior performance of Gemini Pro 2.5 comes at the cost of the longest inference time among MLLMs (18.3 seconds), far exceeding that of other models such as GPT-4o (1.6 seconds). Free360, despite achieving the highest overall performance among enhancement methods, requires 22.5 seconds, approximately 10× longer than its base MLLM (Qwen2.5-VL-7B).

Notably, even with their higher latency, models like Gemini Pro 2.5 and Free360 operate within human response time (28.9 seconds), indicating potential practicality for human-comparable interactive tasks. Moreover, for analysis-driven applications, such as forensic reconstruction, safety auditing, and spatial navigation in complex public environments, comprehensive and accurate scene understanding takes precedence over real-time responsiveness.

The results demonstrate that Free360, through scene-graph-based reasoning and 360-specific operations like entity-centered rotation and view mapping, effectively enhances the 360° image perception of its base MLLM while remaining *practical* in terms of latency.

Inference Time vs. Number of Entities. Figure 7 shows the average inference time versus the number of question-relevant entities detected in Step 1 of Free360. While inference time grows quadratically, this computational cost can be mitigated through parallelized pairwise calls and entity pruning, as discussed in the subsequent subsection.

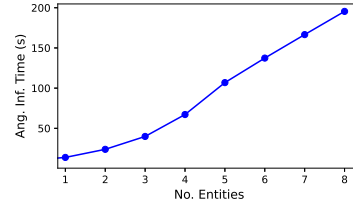


Fig. 7: Average Inference Time vs. Number of Entities.

Potential Strategies for Efficiency Improvement. Free360 consists of modular steps, such as attribute extraction (Step 2) and inter-entity relation detection (Step 3), whose computations over different entities or entity pairs are inherently independent. These operations can therefore be executed in parallel to significantly reduce latency. Additionally, entity identification (Step 1) and attribute extraction (Step 2) can be consolidated into a single model call. While this approach reduces latency to 14.8s, it comes at the cost of degrading accuracy to 42.0%.

Secondly, we may further improve the efficiency through batched inference [8]. Instead of sequentially querying each entity in Step 2 or each pair of entities in Step 3, multiple candidates can be processed within a single forward pass, improving hardware utilization and throughput.

Finally, the detected entities can be ranked and pruned according to their relevance to the query or their detection confidence scores. Prioritizing the most informative candidates reduces unnecessary reasoning steps and further lowers computational cost.

4.6 Qualitative Results

Qualitative results of our Free360 model are provided in Figs. 8–10, covering both successful and failure cases.

Successful Cases. Figure 8, Fig. 9, and Fig. 10(a) show seven examples corresponding to the seven subtasks in 360Bench: FP-IR, FP-IC, PP-IR, PP-IC, SR-Os, SR-OV, and DG, respectively. These examples illustrate how Free360 enhances Qwen2.5-VL-7B’s ability to perceive 360° images by leveraging question-relevant scene graphs. By using the CMP format and image crops to focus on informative regions, Free360 successfully detects entity nodes and their attributes, effectively handling tasks related to fine-grained features, counting, and projection distortion, as shown in Fig. 8 and Fig. 9(a). In Fig. 9(b), inter-entity relations in the scene graph support more accurate spatial reasoning between objects, while Fig. 9(c) and Fig. 10(a) further demonstrate how entity-view relations enable more precise spatial reasoning between objects and the viewer.

Failure Cases. Figures 10(b) and (c) present two representative failure cases. In Fig. 10(b), Free360 struggles with counting several small objects, resulting in an incorrect answer. Figure 10(c) illustrates that the model is unable to generate an accurate routing plan toward the target destination, likely due to the multi-step nature of the task, which complicates reasoning. In this case, although the floating dock appears on the left, reaching it requires turning around and crossing the bridge on the right.

References

1. AframeVR: A-Frame: A web framework for building browser based 3D, AR and VR experiences. <https://github.com/aframevr/aframe> (2025), accessed: 2025-11-05
2. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W., Fu, Z., Xu, Y., Ye, J., Zhang, X., Xie, T., Cheng, Z., Zhang, H., Yang, Z., Xu, H., Lin, J.: Qwen2.5-vl technical report. arXiv preprint arXiv:2502.13923 (2025), <https://arxiv.org/abs/2502.13923>
3. Deitke, M., Clark, C., Lee, S., Tripathi, R., Yang, Y., Park, J.S., Salehi, M., Muenighoff, N., Lo, K., Soldaini, L., Lu, J., Anderson, T., Bransom, E., Ehsani, K., Ngo, H., Chen, Y., Patel, A., Yatskar, M., Callison-Burch, C., Head, A., Hendrix, R., Bastani, F., VanderBilt, E., Lambert, N., Chou, Y., Chheda, A., Sparks, J., Skjonsberg, S., Schmitz, M., Sarnat, A., Bischoff, B., Walsh, P., Newell, C., Wolters, P., Gupta, T., Zeng, K.H., Borchardt, J., Groeneveld, D., Nam, C., Lebrecht, S., Wittlif, C., Schoenick, C., Michel, O., Krishna, R., Weihs, L., Smith, N.A., Hajishirzi, H., Girshick, R., Farhadi, A., Kembhavi, A.: Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. arXiv preprint arXiv:2409.17146 (2024), <https://arxiv.org/abs/2409.17146>
4. Duan, H., Yang, J., Qiao, Y., Fang, X., Chen, L., Liu, Y., Dong, X., Zang, Y., Zhang, P., Wang, J., et al.: Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In: Proceedings of the 32nd ACM International Conference on Multimedia. pp. 11198–11201 (2024)

5. hapePHOTOGRAPHIX: 360° Image. <https://www.flickr.com/photos/hapephotographix/> (2022), credit: hapePHOTOGRAPHIX (<https://www.hapePHOTOGRAPHIX.com>). Accessed: 2025-07-08
6. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021), <https://arxiv.org/abs/2106.09685>
7. Jia, M., Qi, Z., Zhang, S., Zhang, W., Yu, X., He, J., Wang, H., Yi, L.: Omnispatial: Towards comprehensive spatial reasoning benchmark for vision language models. arXiv preprint arXiv:2506.03135 (2026), <https://arxiv.org/abs/2506.03135>
8. Jiang, Y., Irvin, J., Wang, J.H., Chaudhry, M.A., Chen, J.H., Ng, A.Y.: Many-shot in-context learning in multimodal foundation models (2024), <https://arxiv.org/abs/2405.09798>
9. JVET: Algorithm Descriptions of Projection Format Conversion and Video Quality Metrics in 360Lib Version 5 (2017)
10. Laurencon, H., Marafioti, A., Sanh, V., Tronchon, L.: Building and better understanding vision-language models: insights and future directions. arXiv preprint arXiv:2408.12637 (2024), <https://arxiv.org/abs/2408.12637>
11. Laurencon, H., Saulnier, L., Tronchon, L., Bekman, S., Singh, A., Lozhkov, A., Wang, T., Karamcheti, S., Rush, A.M., Kiela, D., Cord, M., Sanh, V.: Obelics: An open web-scale filtered dataset of interleaved image-text documents. arXiv preprint arXiv:2306.16527 (2023), <https://arxiv.org/abs/2306.16527>
12. Laurencon, H., Tronchon, L., Cord, M., Sanh, V.: What matters when building vision-language models? arXiv preprint arXiv:2405.02246 (2024), <https://arxiv.org/abs/2405.02246>
13. Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Zhang, P., Li, Y., Liu, Z., Li, C.: Llava-onevision: Easy visual task transfer (2024), <https://arxiv.org/abs/2408.03326>
14. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. arXiv preprint arXiv:2310.03744 (2024), <https://arxiv.org/abs/2310.03744>
15. Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., Lee, Y.J.: Llava-next: Improved reasoning, ocr, and world knowledge (January 2024), <https://llava-vl.github.io/blog/2024-01-30-llava-next/>
16. Meta: Llama 3.2 model card and prompt formats. https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_2/ (2024), accessed: 2026-03-01
17. NOIRLab: 360 Panorama. <https://noirlab.edu/public/images/archive/category/360pano/> (2025), accessed: 2025-06-20
18. OpenAI: Gpt-4o system card. arXiv preprint arXiv:2410.21276 (2024), <https://arxiv.org/abs/2410.21276>
19. Pichai, S., Hassabis, D., Kavukcuoglu, K.: Introducing Gemini 2.0: our new AI model for the agentic era. <https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024/#gemini-2-0-flash> (2024), accessed: 2025-07-08
20. Shao, H., Qian, S., Xiao, H., Song, G., Zong, Z., Wang, L., Liu, Y., Li, H.: Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. 38th Conference on Neural Information Processing Systems (NeurIPS 2024), Track on Datasets and Benchmarks (2024)
21. Shen, H., Zhao, K., Zhao, T., Xu, R., Zhang, Z., Zhu, M., Yin, J.: Zoomeye: Enhancing multimodal llms with human-like zooming capabilities through tree-based image exploration. arXiv preprint arXiv:2411.16044 (2024)

22. Team, G.: Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multi-modality, Long Context, and Next Generation Agentic Capabilities. arXiv preprint arXiv:2507.06261 (2025), <https://arxiv.org/abs/2507.06261>
23. Team GLM: Chatglm: A family of large language models from glm-130b to glm-4 all tools. arXiv preprint arXiv:2406.12793 (2024), <https://arxiv.org/abs/2406.12793>
24. Wang, S., Pei, M., Sun, L., Deng, C., Li, Y., Shao, K., Tian, Z., Zhang, H., Wang, J.: Spatialviz-bench: A cognitively-grounded benchmark for diagnosing spatial visualization in mllms. arXiv preprint arXiv:2507.07610 (2026), <https://arxiv.org/abs/2507.07610>
25. Wang, W., Lv, Q., Yu, W., Hong, W., Qi, J., Wang, Y., Ji, J., Yang, Z., Zhao, L., Song, X., Xu, J., Xu, B., Li, J., Dong, Y., Ding, M., Tang, J.: Cogvlm: Visual expert for pretrained language models. arXiv preprint arXiv:2311.03079 (2023), <https://arxiv.org/abs/2311.03079>
26. Wang, W., Ding, L., Zeng, M., Zhou, X., Shen, L., Luo, Y., Yu, W., Tao, D.: Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. *Proceedings of the AAAI Conference on Artificial Intelligence* **39**(8), 7907–7915 (2025)
27. Wang, W., Tan, R., Zhu, P., Yang, J., Yang, Z., Wang, L., Kolobov, A., Gao, J., Gong, B.: Site: towards spatial intelligence thorough evaluation. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. pp. 9058–9069 (October 2025)
28. Wang, X., Zhang, X., Luo, Z., Sun, Q., Cui, Y., Wang, J., Zhang, F., Wang, Y., Li, Z., Yu, Q., Zhao, Y., Ao, Y., Min, X., Li, T., Wu, B., Zhao, B., Zhang, B., Wang, L., Liu, G., He, Z., Yang, X., Liu, J., Lin, Y., Huang, T., Wang, Z.: Emu3: Next-token prediction is all you need. arXiv preprint arXiv:2409.18869 (2024), <https://arxiv.org/abs/2409.18869>
29. Wu, H., Huang, X., Chen, Y., Zhang, Y., Wang, Y., Xie, W.: Spatialscore: Towards comprehensive evaluation for spatial intelligence. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2026)
30. Wu, P., Xie, S.: V*: Guided visual search as a core mechanism in multimodal llms. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 13084–13094 (2024)
31. Wu, Z., Chen, X., Pan, Z., Liu, X., Liu, W., Dai, D., Gao, H., Ma, Y., Wu, C., Wang, B., Xie, Z., Wu, Y., Hu, K., Wang, J., Sun, Y., Li, Y., Piao, Y., Guan, K., Liu, A., Xie, X., You, Y., Dong, K., Yu, X., Zhang, H., Zhao, L., Wang, Y., Ruan, C.: Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. arXiv preprint arXiv:2412.10302 (2024), <https://arxiv.org/abs/2412.10302>
32. Xu, G., Jin, P., Wu, Z., Li, H., Song, Y., Sun, L., Yuan, L.: Llava-cot: Let vision language models reason step-by-step. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 2087–2098 (2025)
33. Yang, L., Duan, H., Tao, R., Cheng, J., Wu, S., Li, Y., Liu, J., Min, X., Zhai, G.: Odi-bench: Can mllms understand immersive omnidirectional environments? (2025), <https://arxiv.org/abs/2510.11549>
34. Yang, S., Xu, R., Xie, Y., Yang, S., Li, M., Lin, J., Zhu, C., Chen, X., Duan, H., Yue, X., Lin, D., Wang, T., Pang, J.: Mmsi-bench: A benchmark for multi-image spatial intelligence. In: *ICLR* (2025)
35. Zheng, Z., Yang, M., Hong, J., Zhao, C., Xu, G., Yang, L., Shen, C., Yu, X.: Deepeyes: Incentivizing "thinking with images" via reinforcement learning. arXiv preprint arXiv:2505.14362 (2025), <https://arxiv.org/abs/2505.14362>

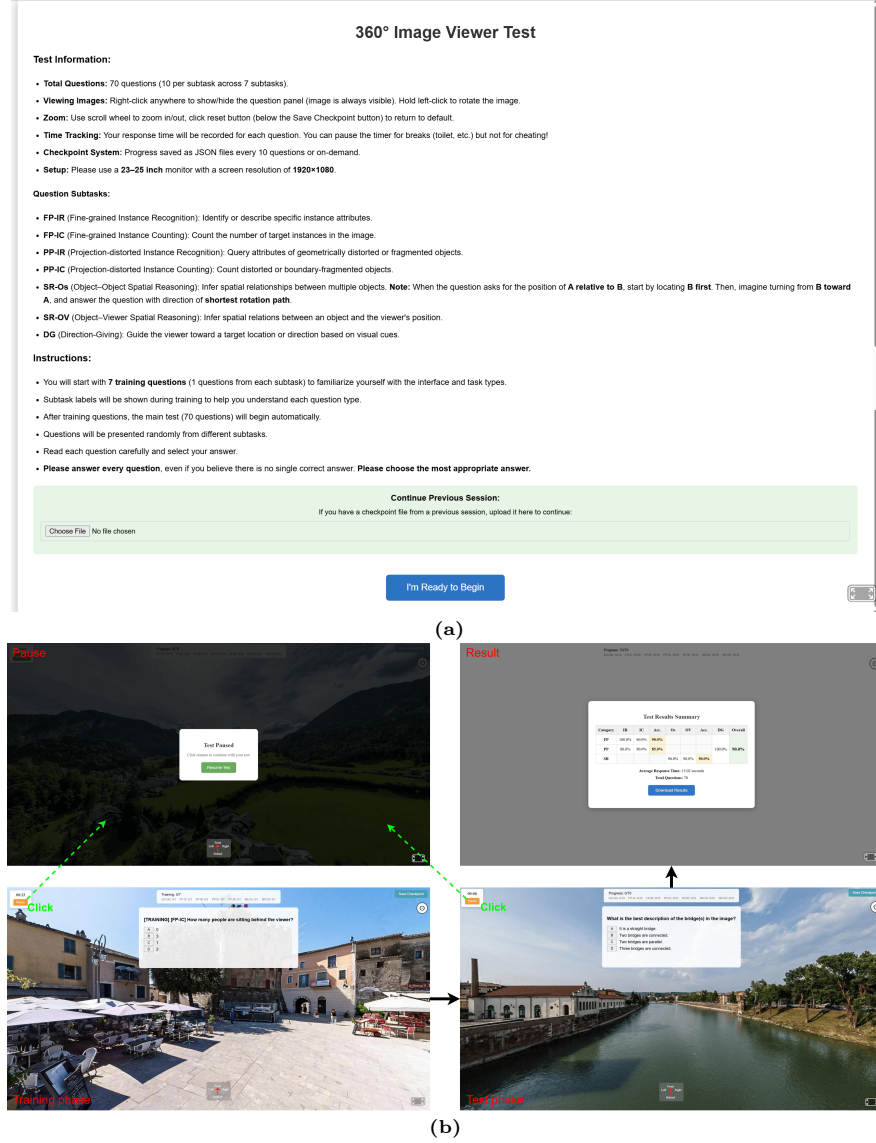


Fig. 6: User interface of our human evaluation study: (a) introduction phase, and (b) training and testing phases. Original 360° image source: [5].

Table 1: Human and model evaluation results. The highest score is highlighted in **bold**. *Acc.* denotes the per-task accuracy, while *Inf. Time* represents the inference time.

Model	Projection format	FP (Acc.)			PP (Acc.)			SR (Acc.)			DG	Overall	Inf. Time (seconds)
		IR	IC	Avg.	IR	IC	Avg.	Os	OV	Avg.	(Acc.)	(Acc.)	
Random guess	—	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	—
Human	—	96.0	82.0	89.0	86.0	78.0	82.0	84.0	90.0	87.0	88.0	86.3	28.9
MLLMs													
GPT-4o [18]	ERP	48.6	32.0	40.3	54.2	38.8	46.5	37.7	35.7	36.7	38.5	40.7	1.5
	CMP	45.5	30.8	38.1	63.0	44.0	53.5	37.4	34.0	35.7	35.3	41.3	1.6
Gemini Flash 2.5 [22]	ERP	62.3	40.0	51.0	44.5	33.8	39.2	37.3	40.3	38.8	40.6	42.7	10.8
	CMP	57.5	41.9	49.6	56.5	46.0	51.3	37.1	25.8	31.5	34.7	42.6	12.5
Gemini Pro 2.5 [22]	ERP	63.2	43.5	53.3	41.1	34.7	37.9	38.0	42.9	40.5	44.0	43.9	18.4
	CMP	61.5	44.8	53.0	52.2	46.7	49.5	46.8	36.9	41.9	37.2	46.5	18.3
Qwen2.5-VL-3B [2]	ERP	56.4	30.6	43.4	38.0	36.4	37.2	27.7	27.9	27.8	30.1	35.2	1.8
	CMP	55.7	28.2	41.8	47.0	43.0	45.0	28.6	28.4	28.5	30.4	37.2	1.9
Qwen2.5-VL-7B [2]	ERP	59.5	27.9	43.6	42.2	36.0	39.1	30.7	31.2	31.0	33.7	37.3	2.1
	CMP	58.1	27.0	42.4	50.6	42.0	46.3	28.4	30.4	29.4	30.6	38.1	2.2
Qwen2.5-VL-32B [2]	ERP	61.0	32.9	46.8	40.4	35.4	37.9	31.6	30.8	31.2	32.3	37.8	4.0
	CMP	57.2	29.6	43.2	54.1	43.2	48.7	34.9	30.2	32.5	29.7	39.7	4.1
DeepSeek-VL2-Small [31]	ERP	41.8	22.7	32.2	42.4	39.3	40.9	27.2	32.6	29.9	32.4	33.9	0.9
	CMP	40.3	19.2	29.7	55.4	43.1	49.2	28.2	29.7	28.9	31.1	35.1	0.9
LLaVA-v1.5-7B [14]	ERP	30.7	8.4	19.4	29.3	35.9	32.6	22.4	26.7	24.5	26.6	25.6	0.4
	CMP	32.5	9.3	20.8	33.4	39.6	36.5	23.5	23.9	23.7	26.3	26.8	0.5
LLaVA-NeXT-7B [15]	CMP	36.8	15.1	25.8	26.1	30.8	28.4	25.0	28.3	26.7	29.4	27.3	0.4
	CMP	34.0	14.3	24.0	33.2	37.1	35.1	24.4	26.9	25.7	27.1	28.0	0.4
LLaVA-OV-0.5B [13]	CMP	36.0	26.2	31.1	34.2	37.9	36.1	23.6	34.1	28.9	31.0	31.8	0.4
	CMP	34.2	26.2	30.2	42.3	33.7	38.0	24.0	30.2	27.1	31.1	31.6	0.6
LLaVA-OV-7B [13]	CMP	53.9	25.7	39.6	32.5	36.5	34.5	28.3	36.1	32.2	40.0	36.1	0.8
	CMP	49.3	26.1	37.6	46.6	41.0	43.8	29.3	31.8	30.6	36.0	37.1	0.9
LLaVA-CoT [32]	ERP	34.9	16.7	25.7	41.5	33.3	37.4	27.2	30.0	28.6	27.5	30.1	2.7
	CMP	33.3	16.9	25.0	55.6	37.3	46.5	24.8	30.9	27.8	27.6	32.2	3.0
Idefics-9B [11]	CMP	30.0	32.9	31.5	29.4	34.7	32.1	25.9	26.6	26.2	28.9	29.8	0.4
	CMP	30.1	31.8	31.0	32.5	35.6	34.0	26.1	24.4	25.3	29.1	29.9	0.4
Idefics2-8B [12]	CMP	37.2	27.9	32.5	28.5	39.8	34.1	31.3	28.6	29.9	25.9	31.3	0.5
	CMP	38.4	27.5	32.9	43.2	39.2	41.2	32.0	27.2	29.6	25.1	33.2	0.6
Idefics3-8B [10]	CMP	41.6	25.7	33.6	33.3	35.4	34.4	34.5	31.7	33.1	31.9	33.4	0.4
	CMP	44.5	25.9	35.1	50.1	41.5	45.8	35.1	28.9	32.0	27.9	36.2	0.5
GLM-4V-9B [23, 25]	CMP	43.8	23.4	33.5	39.0	33.2	36.1	27.3	35.4	31.4	32.2	33.4	0.5
	CMP	42.7	21.8	32.2	52.8	43.9	48.4	28.3	33.3	30.8	31.8	36.2	0.6
Llama-3.2-Vision-11B [16]	CMP	28.4	20.3	24.3	31.4	35.0	33.2	22.0	27.7	24.9	27.2	27.4	0.8
	CMP	30.7	21.5	26.1	47.3	40.5	43.9	22.5	25.6	24.0	27.5	30.7	1.0
Emu3 [28]	CMP	29.4	24.9	27.1	29.6	25.8	27.7	28.9	25.3	27.1	27.5	27.3	0.3
	CMP	29.4	24.9	27.1	29.6	25.8	27.7	28.9	25.3	27.1	27.5	27.3	0.3
Molmo-7B-D [3]	CMP	42.6	29.7	36.1	34.2	32.9	33.5	28.6	41.0	34.9	39.8	35.6	1.2
	CMP	42.3	25.3	33.7	40.1	37.9	39.0	24.4	33.8	29.2	34.5	34.0	1.2
Molmo-7B-O [3]	CMP	42.4	30.9	36.6	37.3	38.8	38.1	25.5	38.4	32.0	37.7	35.8	1.1
	CMP	41.5	26.9	34.1	53.3	42.0	47.7	23.8	29.1	26.5	32.3	35.4	1.1
MLLM Enhancement Models													
SEAL [30]	ERP	31.3	24.6	27.9	30.4	27.8	29.1	23.6	20.4	22.0	18.9	25.2	14.6
	CMP	33.6	21.4	27.5	32.1	30.4	31.3	23.6	18.7	21.1	20.7	25.7	14.6
DC ² [26]	ERP	40.0	16.0	27.9	44.0	36.4	40.2	22.0	25.1	23.6	21.5	29.1	617.9
	CMP	39.7	11.0	25.2	49.1	34.9	42.0	20.2	23.1	21.6	20.2	28.1	761.6
ZoomEye [21]	ERP	61.5	26.6	43.9	44.2	31.9	38.1	25.3	31.3	28.3	28.5	35.5	19.5
	CMP	59.2	22.8	40.8	51.2	30.3	40.7	27.3	29.1	28.2	28.0	35.3	23.9
VisCoT-7B-336 [20]	ERP	28.1	23.3	25.7	30.6	35.9	33.3	22.3	26.7	24.4	23.8	27.2	1.1
	CMP	28.3	23.7	26.0	34.8	36.4	35.6	24.6	24.4	24.5	24.2	28.0	0.8
DeepEyes [35]	ERP	53.6	25.2	39.3	33.3	32.0	32.7	28.2	26.9	27.5	29.1	32.6	78.4
	CMP	52.6	25.6	39.0	43.5	42.5	43.0	31.4	26.3	28.9	27.3	35.5	66.9
Omni-CoT [33]	Hybrid	58.9	24.3	41.4	49.5	40.3	44.9	28.3	41.0	34.7	35.0	39.5	16.6
Free360 (Ours)	Hybrid	60.7	30.0	45.2	54.7	43.9	49.3	33.6	53.3	43.6	41.4	45.3	22.5

Table 2: Prompt templates for identifying question-relevant entities (Step 1) and extracting attributes (Step 2).

Role	Prompt Template
Identifying question-relevant entities	You are given a 360-degree image and a question about that image. Your task is to determine whether any entity mentioned in the question is visibly present in the given image.
	Follow these steps carefully: 1. Read the question, identify all relevant entities mentioned in the question. 2. Outline the position of all relevant entities and output their bounding box coordinates in the format [x1, y1, x2, y2] (top-left and bottom-right corners), along with the entity label
	Return your response strictly in the following JSON format: <pre>{ "analysis": "<list of relevant entities>", "coordinates": [{ "bbox_2d": [x1,y1,x2,y2], "label": <entity_label> }, { "bbox_2d": [x1,y1,x2,y2], "label": <entity_label> }, ...] }</pre>
Extracting attributes	Now, process the given input: Question: {Question} Output:
	You are given an image crop extracted from a 360-degree image. Your task is to provide a detailed description of the {Entity Label} visible in the crop to help answer the question: {Question}. If any text is visible, transcribe it completely. Do not guess or describe anything that cannot be clearly seen.
	Return your response strictly in the following JSON format: <pre>{ "Description": "<detailed description in natural language>", "Text": [<List of texts>] }</pre> Output:

Table 3: Prompt template for detecting inter-entity relations (Step 3).

Role	Prompt Template
Detecting inter-entity relations	You are given a 360-degree image and a question about that image. In the image, two entities are highlighted using colored bounding boxes, each with an associated label and boundary color. Your task is to describe the spatial relationship between these entities from the observer’s perspective, to help answer the question. Follow these steps carefully: 1. Use the boundary colors to locate each labeled entity in the image. If any bounding box is not visible or if the spatial relationship cannot be determined, return an empty string “”. 2. Analyze the spatial relationship between the visible entities from the observer’s perspective to help answer the question. Do not guess or describe anything that cannot be clearly seen. 3. Describe the spatial relationship in details to help answer the question USING the entity labels in the format: Label1 ->spatial relationship ->Label2. Return your response strictly in the following JSON format: <pre>{ "analysis": "<Your reasoning based on bounding boxes>", "output": "<Label1 ->spatial relationship ->Label2>" }</pre> Now, process the given input: Question: {Question} Entity labels: {Entity Labels} Boundary colors: {Legend} Output:

Table 4: Prompt template for final question answering based on scene graphs.

Role	Prompt Template
Final question answering based on scene graphs	You are given a question about a 360-degree image and a scene graph describing the entities in the image, their attributes, the spatial relations between them, and the spatial relations between them and the viewer. Your task is to answer the question by analyzing the scene graph. Follow these steps carefully: 1. Analyze the scene graph: Analyze the attribute edges and spatial relation edges to find nodes relevant to the question. 2. Answer: Select the best answer from the multiple-choice options (A, B, C, or D). If the scene graph does not provide enough information to answer the question, answer with “CANNOT ANSWER”. Return your response strictly in the following JSON format: <pre>{ "analysis": "<Your reasoning about scene graph>", "answer": "<Only 'A', 'B', 'C', 'D' OR 'CANNOT ANSWER'>" }</pre> {In-context Examples} Now, process the given input: Question: {Question} Scene graph: {Scene Graph} Output:



Free360

List of Nodes: man 1, red umbrella 1, behind view

Attribute relations:

- man 1 →he is wearing a light-colored jacket, dark pants, and **white sneakers**. He is holding a red umbrella in his left hand....
-

Spatial relations: man 1 → in → behind view,...

Generated scene graph

Answer of Models

GPT-4o	B	DC ²	–
Gemini Pro 2.5	B	ZoomEye	B
Qwen2.5-VL-7B	B	VisCoT-7B-336	A
DeepSeek-VL2-Small	B	DeepEyes	A
LLaVa-CoT	B	Omni-CoT	A
SEAL	A	Free360 (Ours)	A

What is the color of the shoes worn by the man holding a red umbrella?

A. **White**
 B. Black
 C. Brown
 D. Grey

(a)



Free360

List of Nodes: flags 1, flags 2, front view

Attribute relations:

- flags 1 → ...appear to be rectangular...
- flags 2 → ...flagpole has a flag attached...
- front view → front of the scene and front of the viewer

Spatial relations: flags 1 → in → front view, flags 2 → in → front view

Generated scene graph

Answer of Models

GPT-4o	B	DC ²	B
Gemini Pro 2.5	B	ZoomEye	C
Qwen2.5-VL-7B	C	VisCoT-7B-336	A
DeepSeek-VL2-Small	C	DeepEyes	B
LLaVa-CoT	A	Omni-CoT	C
SEAL	D	Free360 (Ours)	B

How many flags are mounted on top of Rita Hillenbrand Stadium?

A. 0
 B. **2**
 C. 1
 D. 3

(b)



Free360

List of Nodes: bridge 1, bottom view

Attribute relations:

- bridge 1 → **a wooden bridge spanning across a river** in a picturesque setting....
- bottom view → bottom of the scene and below the viewer

Spatial relations: bridge 1 → in → bottom view

Generated scene graph

Answer of Models

GPT-4o	C	DC ²	C
Gemini Pro 2.5	B	ZoomEye	A
Qwen2.5-VL-7B	D	VisCoT-7B-336	A
DeepSeek-VL2-Small	A	DeepEyes	D
LLaVa-CoT	C	Omni-CoT	D
SEAL	A	Free360 (Ours)	C

What can you tell about the bridge(s) in the image?

A. There are two bridges connected to each other.
 B. There are three bridges merged at the center.
 C. **It is a straight bridge that crosses over one river.**
 D. It is a curved bridge that crosses over two rivers.'

(c)

Fig. 8: Predictions from Free360 (Ours) and other models on 360Bench samples. Correct answers are highlighted in red. All models take 360° CMP images as input. “–” indicates cases where the model could not produce a valid answer. Question-relevant information in the scene graphs produced by Free360 is highlighted in green. Original 360° image sources: [5, 17].

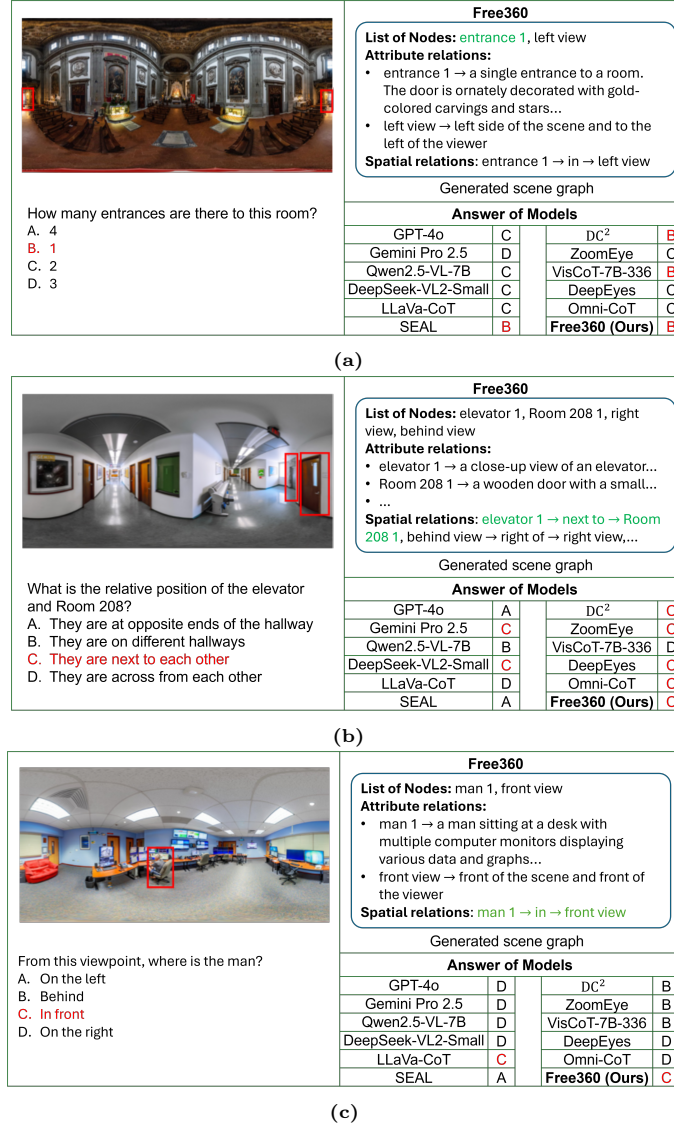


Fig. 9: Predictions from Free360 (Ours) and other models on 360Bench samples. Correct answers are highlighted in red. All models take 360° CMP images as input. Question-relevant information in the scene graphs produced by Free360 is highlighted in green. Original 360° image sources: [5, 17].

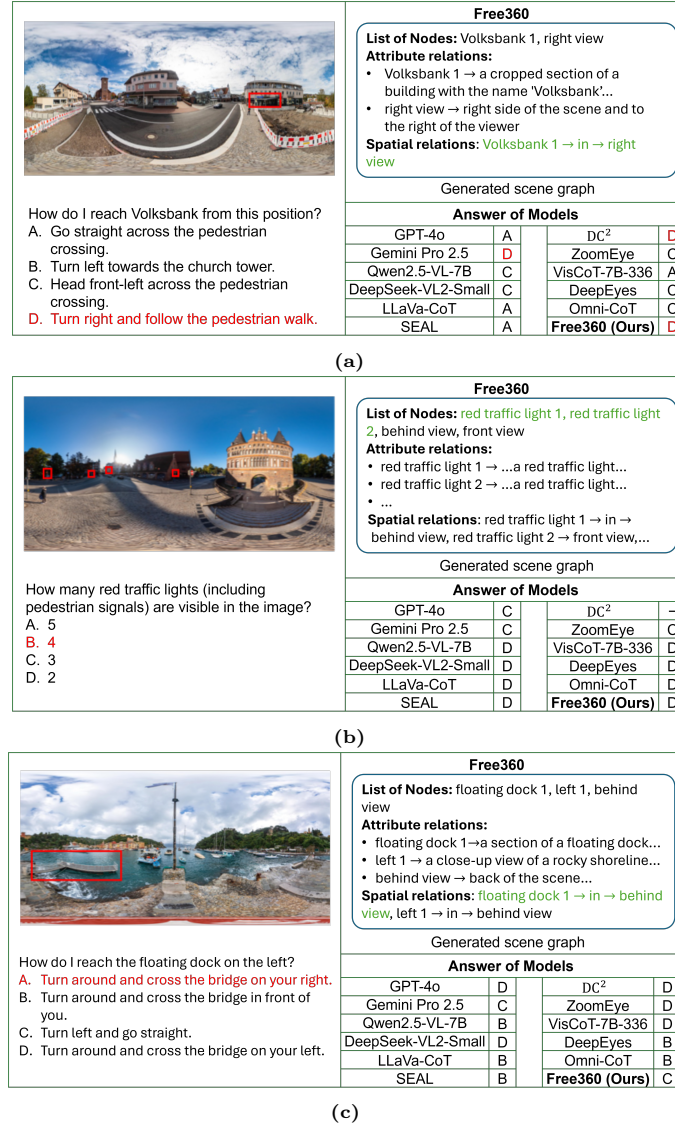


Fig. 10: Predictions from Free360 (Ours) and other models on 360Bench samples. Correct answers are highlighted in red. (a) depicts a successful case, while (b)–(c) illustrate failure cases. All models take 360° CMP images as input. “–” indicates cases where the model could not produce a valid answer. Question-relevant information in the scene graphs produced by Free360 is highlighted in green. Original 360° image sources: [5].