TRAJSCOGMAP-BENCH: EVALUATING COGNITIVE MAPPING AND SPATIAL MEMORY FROM EGOCENTRIC VIDEO TRAJECTORIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Navigation requires more than remembering visited places: agents must integrate egocentric observations into spatial knowledge that supports subsequent queries and route planning. Evaluating this ability requires distinguishing local spatial understanding, integration across trajectories, and the retention of spatial information for future use. We introduce **TrajsCogMap-Bench**, an open-loop benchmark for evaluating cognitive mapping and spatial memory in multimodal large language models. Built from real-world egocentric videos in indoor environments rich in points of interest, the benchmark organizes spatial queries and route-prediction tasks across three levels: local motion understanding, single-trajectory integration, and multi-trajectory reasoning. The multi-trajectory setting probes spatial knowledge construction and use from multiple traversals of the same environment, covering different locations and viewpoints, including route inference beyond directly experienced paths. We report answer accuracy for all tasks except open-ended route navigation. For the latter, an executor model follows the generated instructions in a simulator grounded in the environment topology and egocentric observations, with performance measured by success rate (SR) and success weighted by path length (SPL). We provide complementary full-context and online-memory protocols: full-context evaluation tests joint integration of observations available at answering time, while online-memory evaluation tests the use of memory built incrementally from sequential observations without knowledge of future questions. Our evaluations reveal a gap between remembering landmarks and reasoning about their spatial relations to plan routes. Performance further declines when models rely on incrementally constructed memory, highlighting limitations in how current memory systems preserve spatial information for subsequent reasoning.

1 INTRODUCTION

Multimodal large language models (MLLMs) have emerged as powerful perceptual and reasoning backbones for modern embodied agents. By unifying visual perception, language comprehension, and general-purpose reasoning, MLLMs achieve strong performance on image recognition, video question answering, visual instruction following, and interactive decision-making tasks. Nevertheless, real-world mobile navigation demands more than isolated visual recognition and short-video reasoning. To operate reliably in complex public environments, a mobile agent must continuously interpret dynamic egocentric observations and estimate its self-motion. It also needs to track its orientation and displacement over time, associate landmarks across different viewpoints, and reason about geometric distances, topological connections, and navigable paths. Such navigation therefore depends on persistent spatial memory that supports long-horizon spatial reasoning beyond immediate perception. When navigating shopping malls, parking garages, or large-scale public spaces, agents are required to answer complex spatial queries regarding past landmark occurrences, relative distances, and feasible navigation routes. Addressing these tasks requires the model or agent to memorize spatial and temporal landmark information, infer cross-view geometric relationships, and assemble fragmented visual observations into coherent environmental layouts. An agent may encounter different parts of the same environment on separate traversals, observing different locations

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

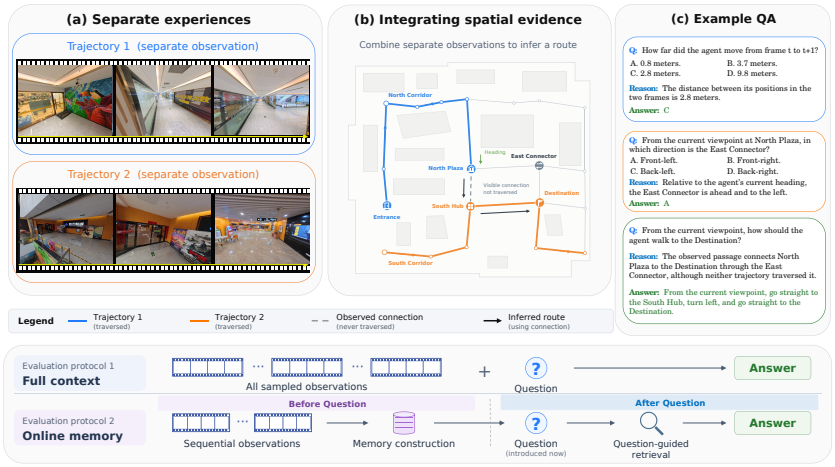


Figure 1: Task overview and evaluation protocols for spatial reasoning across separate experiences. (a) Two separately observed trajectories provide complementary visual evidence about a shared environment. (b) Integrating this evidence reveals spatial connections that were observed but not traversed, enabling route inference across trajectories. Colored arrows indicate traversed paths, gray dashed lines denote observed but untraversed connections, and black arrows show an inferred route. The spatial schematic is for illustration only and is not provided to the model. (c) Example questions cover displacement estimation, egocentric direction recognition, and route planning, with illustrative reasoning and answers. Bottom: In the full-context protocol, all sampled observations are provided together with the question. In the online-memory protocol, observations are processed sequentially to construct memory before the question is revealed; the question then guides memory retrieval for answering.

and viewpoints each time. Building useful spatial knowledge requires relating these experiences through shared landmarks and spatial connections. Such challenges become particularly pronounced when critical spatial cues are sparsely distributed across long or temporally separated trajectory segments. This requires transforming unstructured egocentric visual streams into stable, structured, and queryable spatial memory.

Recent navigation methods use topological maps and dynamic memory to support language-guided spatial reasoning and planning Chen et al. (2024); Zhang et al. (2025). Complementary benchmarks evaluate the underlying abilities: VSI-Bench, SpatialBench, and Video2Mental examine spatial understanding and mental navigation Zhang et al. (2024); Zheng et al. (2025); Wang et al. (2026b), while UCS-Bench and GST-Bench probe continual spatial memory and global spatial awareness from egocentric video Zhou et al. (2026); Yang et al. (2026). Cross-video evidence integration is also studied by EgoExoMem through synchronized views and by EgoMonth through temporally separated recordings Lee et al. (2026); Yin et al. (2026). These advances motivate a more specific evaluation of how models combine spatially connected trajectories to infer routes beyond those directly experienced. We pair this evaluation with full-context and question-independent online-memory protocols on the same questions, extending capability-oriented navigation evaluation and embodied memory assessment Li et al. (2025b); Xu et al. (2026). This comparison characterizes the effects of relying on constructed memory, although task accuracy alone does not uniquely identify the causes of failure.

To address this gap, we introduce **TrajsCogMap-Bench**, an open-loop diagnostic benchmark for evaluating cognitive mapping and spatial memory in multimodal large language models. Built from real-world egocentric videos in POI-rich indoor environments, TrajsCogMap-Bench uses points of interest as persistent spatial anchors for probing how models organize, retain, and reuse spatial information. To enable fine-grained diagnosis, TrajsCogMap-Bench structures its question-answering tasks into three progressive levels, from foundational local motion understanding to single-trajectory integration and multi-trajectory spatial memory reasoning. These tasks assess local motion under-

standing and spatial integration within and across trajectories, including the ability to infer routes beyond those directly experienced.

We further design two complementary evaluation protocols to systematically examine MLLM-based spatial reasoning and memory utilization. The full-context protocol provides all sampled observations in chronological order together with the question, testing whether models can jointly integrate the available experiences to answer spatial queries. The online-memory protocol instead processes observations sequentially to construct and update question-independent memory; after observation is complete, models answer spatial queries using the resulting memory and question-specific inputs. Together, the protocols evaluate spatial knowledge use with joint access to observations and with incrementally constructed memory. All evaluations follow an open-loop setting with fixed observation trajectories, avoiding confounding effects from differences in exploration behavior across models and memory mechanisms. We instantiate two memory configurations: text-only memory inspired by ReMEmbR that stores textual descriptions of observations, and image-text memory inspired by RAVEN that retains both visual frames and text descriptions. Our evaluations reveal a gap between remembering landmarks and reasoning about their spatial relations to plan routes. Performance further declines under the online-memory protocol, suggesting that the evaluated memory systems do not consistently retain or retrieve the spatial information needed for subsequent reasoning.

Our main contributions are summarized as follows:

- We establish a hierarchical cognitive-mapping evaluation framework that organizes spatial QA tasks from egocentric video trajectories into three progressive levels: local motion understanding, single-trajectory integration, and multi-trajectory reasoning with topology-grounded route inference.
- We construct **TrajsCogMap-Bench**, an open-loop multimodal QA benchmark built from real-world POI-rich indoor egocentric videos, with topological, metric, floor-level, and POI/landmark annotations enabling programmatically verifiable spatial and route-inference evaluation.
- We introduce complementary full-context and online-memory protocols with two memory settings—text-only and image-text—to evaluate joint integration of observations and the use of incrementally constructed, question-independent memory for subsequent spatial queries.

2 Related Work

MLLM-Based Navigation and Topological Mapping. Vision-and-language navigation evaluates instruction following in visual environments, with representative benchmarks including R2R Anderson et al. (2018b) and VLN-CE Krantz et al. (2020). MapGPT uses online topological maps for prompted path planning, while MSNav combines dynamic memory with LLM spatial reasoning for zero-shot navigation Chen et al. (2024); Zhang et al. (2025). DreamNav studies trajectory-based imaginative planning in continuous environments, and Topo-AeroVLN studies cognitive topological mapping for aerial navigation Li et al. (2025a); Wang et al. (2026a). Related approaches develop adaptive graph granularity, topology- and visitation-aware waypoint prompting, and joint metric–topological representations Ding et al. (2026); Wang et al. (2025); Zhou et al. (2025). These methods motivate evaluating the spatial knowledge used in navigation. Alongside capability-oriented evaluations such as CapNav Li et al. (2025b), our benchmark probes this knowledge using fixed observations, so model comparisons do not depend on different exploration policies.

Spatial and Streaming Video Benchmarks. VSI-Bench, SpatialBench, and Video2Mental evaluate spatial reasoning and cognitive mapping from visual observations Zhang et al. (2024); Zheng et al. (2025); Wang et al. (2026b). GST-Bench examines global spatial awareness beyond current-view perception, while REM probes cognitive mapping from synthetic egocentric trajectories Yang et al. (2026); Smith et al. (2025). UCS-Bench evaluates continual orientation, trajectory, and reachability memory; OVO-S-Bench evaluates hierarchical spatial intelligence from video prefixes; and SpaMEM separates perception, state-history reasoning, and spatial belief maintenance under scene changes Zhou et al. (2026); Bohacek et al. (2026); Teney et al. (2026). S^3-Bench further combines streaming spatial reasoning with active exploration Liu et al. (2026). Our evaluation complements

these settings with spatially related trajectory groups and topology-grounded route inference, assessed under both full-context and question-independent online-memory protocols with fixed observations.

Memory-Augmented Embodied Models. MultiON benchmarks semantic map memory through sequential object navigation Gordon et al. (2020), and 3DLLM-Mem retrieves and fuses episodic spatial observations for long-horizon embodied reasoning Huang et al. (2025). LMEE studies memory for exploration and subsequent questions, while VL-KnG constructs persistent spatiotemporal knowledge graphs Xu et al. (2026); Chen et al. (2025). ReMEmbR stores textual descriptions for retrieval and navigation, whereas RAVEN retains visual evidence in visuo-spatio-temporal memory Anwar et al. (2024); Hu et al. (2026); these approaches inspire our text and image–text configurations. We compare these configurations with full-context access on the same spatial questions. Because representations, retrieval mechanisms, and budgets differ, the comparison measures complete systems rather than the isolated effect of memory structure.

3 TRAJSCOGMAP-BENCH

3.1 OVERVIEW

TrajsCogMap-Bench evaluates how MLLMs integrate egocentric observations into spatial knowledge that supports subsequent queries and route inference. Our benchmark focuses on POI-rich indoor environments, such as shopping malls and commercial complexes, where POIs serve as persistent spatial anchors for associating observations across trajectories and reasoning about navigable routes. These environments feature visually similar corridors and intersections, branching walkways, and layouts spanning multiple floors or zones, posing challenges for landmark association and spatial integration.

For each environment, we collect panoramic videos along recorded trajectories and render egocentric views from them. We also annotate the environment's topological structure and spatial relations, using these annotations to construct spatial questions and derive their reference answers. We organize the tasks into three levels according to the scope of the provided observations. Local motion understanding tests whether models can infer egocentric motion from short observation sequences. Single-trajectory integration evaluates whether models can combine observations along one trajectory to recall POIs and reason about their order and spatial relations. Multi-trajectory reasoning evaluates whether models can build and use spatial knowledge from multiple traversals of the same environment, each covering different locations and viewpoints. Its tasks probe landmark recall, spatial relations, and route inference using the accumulated observations, including routes that connect separately experienced segments or exploit visible shortcuts.

We evaluate the same QA instances under two complementary protocols: full-context and online-memory. In the full-context protocol, all sampled observations from the associated trajectory or trajectory group are jointly available with the question, allowing spatial information to be integrated at answering time. In the online-memory protocol, models process the same observations sequentially and incrementally construct memory without access to future questions; after all observations have been processed, they answer questions using the resulting memory and question-specific inputs. This comparison assesses spatial knowledge use under joint observation access and sequential memory construction.

3.2 BENCHMARK CONSTRUCTION

Data Collection and Pose Estimation. We recorded panoramic videos using an Insta360 camera in POI-rich indoor environments, including shopping malls and commercial complexes. Each environment asset corresponds to a distinct location, with no spatial overlap between assets. Camera poses were estimated using simultaneous localization and mapping (SLAM) to recover the recorded trajectories and provide a spatial reference for manual annotation.

Topological and Spatial Annotation. For each environment, we manually annotate POI locations along the reconstructed trajectories and construct a scene-level topological map using the trajectories and corresponding panoramic observations. Each position is associated with its corresponding

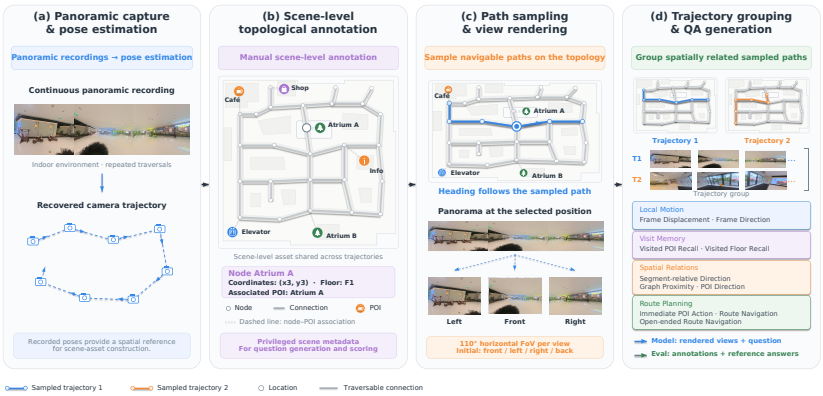


Figure 2: Construction pipeline of TrajsCogMap-Bench. (A) Panoramic recordings and estimated camera poses provide the basis for scene reconstruction. (B) Manual annotation produces a scene-level topological map with navigable nodes, traversable connections, and POI annotations. (C) Paths are sampled on the annotated map, and path-dependent headings guide the rendering of egocentric perspective views. (D) Spatially related trajectories are grouped for QA generation. Models receive the rendered observations and questions, while the full reference topology is used for answer generation and scoring. Memory information access is detailed in Appendix C.1.

panoramic observation. The map represents navigable locations, such as intersections and passage entrances, as nodes and direct traversable connections between them as edges. Each node is annotated with map coordinates, a floor label, and associated POIs, including storefronts, service counters, escalators, and elevator halls. The resulting map provides a shared spatial reference for question generation and automatic scoring.

Trajectory Grouping and View Rendering. We manually select multiple trajectories from the same location to form a multi-trajectory group. Group selection considers shared POIs, overlapping paths, or visible spatial connections that link the trajectories, together with complementary coverage of different local regions for spatial questions and cross-trajectory route inference. For cases involving untraversed connections, annotators verify that the relevant connection is visible in the provided views and traversable, but is not traversed by any of the selected trajectories. We then render egocentric observations with viewing headings determined by the direction of motion along each trajectory. From each panorama, we generate front, left, right, and back perspective views with a horizontal field of view of 110°. The rendered views are ordered along each annotated trajectory to form an egocentric observation sequence. The initial observation includes front, left, right, and back views; subsequent observations include front, left, and right views.

Task Taxonomy. We organize tasks by the scope of the provided observations: local motion understanding, single-trajectory integration, and multi-trajectory reasoning. Local motion tasks assess displacement and heading changes between adjacent observations. The single- and multi-trajectory settings share the same task families, including POI and floor recall, route description, graph proximity, POI-relative direction, and POI-directed route prediction. They differ in whether a question is answered using observations from one trajectory or a group of trajectories within the same environment. This design evaluates spatial knowledge use after one traversal or multiple traversals covering different parts and viewpoints of the same environment. Recall tasks probe retained landmark and floor information, while spatial-relation and route tasks probe how the available observations support spatial judgments and planning. The observation scope defines the setting; individual questions may draw on different subsets of the supplied evidence. Detailed task definitions and reference-answer generation procedures are provided in Appendix A.2.

QA Generation and Ground Truth. We use task-specific generators to construct questions and reference answers from the sampled trajectories and their spatial annotations. For each question, the generator selects the target POIs, locations, or trajectory intervals and instantiates the correspond-

Table 1: Question counts across the nine tasks. Percentages use all 1,921 questions as the denominator and may not sum to 100% due to rounding.

Task	Questions	Share (%)
Route Navigation	387	20.15
Open Route Navigation	387	20.15
Frame Direction	215	11.19
POI Direction	212	11.04
Immediate POI Action	212	11.04
Frame Displacement	208	10.83
Graph Proximity	174	9.06
Visited POI Recall	83	4.32
Visited Floor Recall	43	2.24
Total	1,921	100.00

ing question template. POI recall answers are derived from whether the trajectory passes through the corresponding annotated POI nodes; seeing a POI from another node does not count as a visit. Visited-floor answers are obtained from the floor labels of the traversed nodes. Displacement and heading-change answers are computed from the positions and viewing headings of the selected observations. POI-relative directions are calculated using POI coordinates and the reference orientation specified in the question. Turn sequences and reference routes are derived from the ordered trajectory nodes and their associated headings. Graph-proximity and route-planning answers are computed using the annotated connectivity and the task-specific constraints. All generated QA pairs undergo manual review. Quality control excludes samples with missing required images, multiple valid answer options, or numerical options that are too close to distinguish reliably. Format and duplicate checks use the associated trajectories and question text. Questions use multiple-choice, true/false, or open-ended formats. Open-ended responses are used for route instructions.

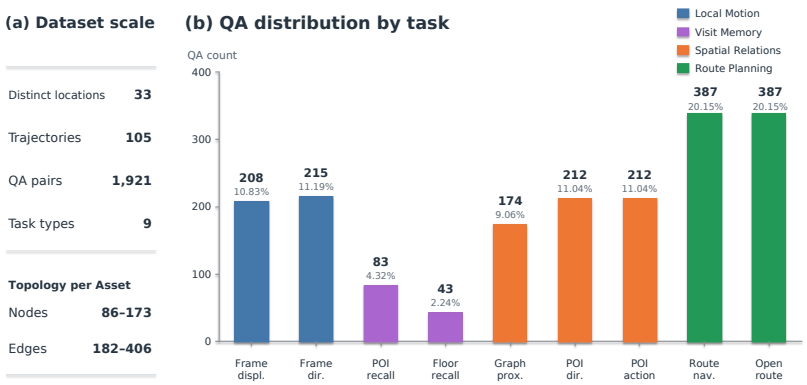


Figure 3: Dataset scale and task distribution of TrajsCogMap-Bench. (a) The benchmark comprises 105 trajectories from 33 spatially non-overlapping locations and 1,921 QA pairs across nine task types. Node and edge counts indicate the range of annotated topology sizes per location. (b) QA counts by task, with colors denoting the four task families.

Dataset Statistics. Figure 3 summarizes the scale and composition of TrajsCogMap-Bench. The benchmark contains 105 trajectories collected from 33 distinct, spatially non-overlapping locations and 1,921 QA pairs across nine task types. Each location's annotated topology contains 86–173 nodes and 182–406 edges.

Table 1 gives the per-task counts. Full-context overall answer accuracy pools 1,534 questions, excluding the 387 open-route questions. Memory evaluation additionally excludes the 423 local-

motion questions, leaving 1,111 questions for overall answer accuracy and 387 open-route questions scored separately.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Models. The full-context evaluation in Table 2 includes three API-based models—Qwen3.8-Max, Kimi K3, and Claude 5 Sonnet—and four open-weight models: Qwen3-VL at 2B, 4B, 8B, and 32B. The memory comparison in Table 3 uses Qwen3.8-Max as the backbone, with a full-context reference on the same questions. Model evaluation scope, input conventions, and configuration details are listed in Appendix B.

Protocols. We compare full-context integration and online-memory construction using the same QA instances and sampled observations. For tasks requiring current-view images, both protocols provide identical current views. In the *full-context* protocol, models receive the question together with all associated observations, ordered along each trajectory with explicit trajectory boundaries. In the *online-memory* protocol, models process observations sequentially and construct external memory before the questions are revealed. At answering time, they use this memory, retrieved evidence, and question-specific inputs without receiving the full observation sequences again. We evaluate two online-memory configurations:

- *Text memory* stores per-step textual descriptions and trajectory-level summaries, which are integrated into a spatial summary. At answering time, the model uses these summaries and retrieves relevant textual records. Historical images are not retained.
- *Image–text memory* augments the textual memory with stored visual frames. At answering time, the model retrieves relevant text and images to supplement the summaries.

Memory construction, information access, retrieval tools, and configuration details are provided in Appendix C.1.

Evaluation Metrics. For all tasks except open-ended route navigation, we report answer accuracy using task-specific answer matching. Overall accuracy is pooled across these tasks, excluding open-ended route navigation. For open-ended route navigation, we use Qwen3.8-Max as the shared navigation executor for all evaluated models, using the same execution prompt. The executor receives the evaluated model's generated route instructions and follows them by selecting actions based on its current egocentric observations in a simulator grounded in the annotated environment topology. We evaluate the resulting navigation trajectories using success rate (SR) and success weighted by path length (SPL) Anderson et al. (2018a), measuring goal-reaching success and navigation efficiency, respectively. These execution-based metrics are reported separately from answer accuracy and reflect both instruction quality and executor performance. Executor configuration and input details are listed in Appendix B.1; scoring is defined in Appendix F.

4.2 MAIN RESULTS

Model scale and spatial task performance. Within the Qwen3-VL family, overall answer accuracy increases from 33.1% at 2B to 36.4% at 4B, 38.9% at 8B, and 43.6% at 32B. All eight answer-accuracy tasks improve across these model sizes. Route-choice accuracy rises from 25.1% to 41.1%, and immediate POI-action accuracy rises from 40.1% to 55.2%. Directional tasks remain difficult: POI-direction accuracy increases from 5.2% to 11.8%, while frame-direction accuracy increases from 14.0% to 18.1%. Thus, the evaluated larger models perform better, but substantial errors remain even with access to the full observation context.

Landmark recall and spatial relations. Visited-POI recall ranges from 85.5% to 94.0% across the Qwen3-VL models, alongside lower accuracy on direction and route tasks. For example, the 32B model achieves 94.0% POI recall, 11.8% POI-direction accuracy, and 41.1% route-choice accuracy. These results show that high recall scores can coexist with frequent errors on spatial queries. Because recall is binary and the other tasks differ in answer format and difficulty, their raw accuracy

Table 2: Full-context evaluation results on TrajsCogMap-Bench. Accuracy and SR are reported as percentages; SPL is reported on a 0–1 scale. Tasks use answer accuracy except open-ended route navigation, which uses SR and SPL. Overall accuracy pools the eight answer-accuracy tasks and excludes open-ended route navigation. Overall accuracies for the open-weight models and Claude 5 Sonnet are computed from the supplied correct-answer counts over 1,534 questions. Qwen3.8-Max and Kimi K3 overall values remain provisional weighted estimates pending verification of exact counts. GT instructions report reference-instruction execution. Dashes denote inapplicable metrics.

	Local Motion		Visit Memory		Spatial Relations		Routes				
Model	Frame displ.	Frame dir.	POI recall	Floor recall	Graph prox.	POI dir.	Route choice	POI action	Open SR	Open SPL	Overall Acc.
API-based Models											
Qwen3.8-Max	81.3	82.8	92.8	83.7	79.9	58.5	83.5	64.2	31.3	0.307	77.1
Kimi K3	64.9	70.7	86.7	74.4	67.2	40.6	68.2	47.2	24.5	0.235	62.5
Claude 5 Sonnet	73.1	64.2	94.0	65.1	77.0	50.0	78.3	48.6	28.4	0.278	67.9
Open-weight Models											
Qwen3-VL-2B	52.4	14.0	85.5	11.6	56.9	5.2	25.1	40.1	11.1	0.092	33.1
Qwen3-VL-4B	54.8	15.3	90.4	20.9	59.8	6.1	30.0	44.8	13.4	0.115	36.4
Qwen3-VL-8B	56.3	16.3	91.6	25.6	62.1	7.1	34.1	48.1	14.2	0.102	38.9
Qwen3-VL-32B	58.2	18.1	94.0	34.9	66.1	11.8	41.1	55.2	16.8	0.135	43.6
Reference-instruction execution											
GT instructions	—	—	—	—	—	—	—	—	70.3	0.586	—

differences do not isolate a memory-versus-reasoning effect. Perception, viewpoint association, and spatial reasoning may all contribute to the observed errors.

Route selection and instruction execution. Route-choice accuracy and open-route SR measure different outcomes. The former evaluates selection among candidate instructions; the latter evaluates whether generated instructions lead the shared executor to stop at the target. Qwen3.8-Max obtains 83.5% route-choice accuracy and 31.3% execution SR, while Qwen3-VL-2B obtains 25.1% and 11.1%, respectively. These values should not be interpreted as a direct conversion between the tasks. Reference instructions achieve 70.3% execution SR, showing that execution can fail even with reference guidance. Open-route scores therefore reflect both instruction generation and executor behavior.

We next examine how performance changes when direct access to the observation sequences is replaced by incrementally constructed external memory.

4.3 Memory Evaluation

Table 3 compares text and image–text memory using Qwen3.8-Max as the backbone model. The comparison includes all evaluation questions except the two local motion tasks, Frame Displacement and Frame Direction. The full-context reference and both memory configurations are evaluated on exactly these same questions.

For Qwen3.8-Max, route-choice accuracy decreases from 83.5% with full-context access to 64.1% with text memory and 62.8% with image–text memory. Open-route SPL decreases from 0.307 to 0.105 and 0.113, respectively. These results characterize the evaluated memory systems with this backbone; they do not establish that the same effects hold across other backbone models.

5 Conclusion

We introduced TrajsCogMap-Bench, a benchmark for evaluating cognitive mapping and spatial memory from real-world egocentric observations. Its tasks span local motion understanding, single-trajectory integration, and multi-trajectory reasoning, including route inference beyond directly experienced paths. Complementary full-context and online-memory protocols examine spatial knowledge use through joint integration of available observations and through memory incrementally constructed from sequential observations before questions are revealed. Our evaluations reveal a gap

Table 3: Memory evaluation on TrajsCogMap-Bench. Qwen3.8-Max is evaluated under the full-context protocol and two online-memory configurations on all evaluation questions except Frame Displacement and Frame Direction. Accuracy and SR are reported as percentages; SPL is reported on a 0–1 scale. All tasks except open route narration use answer accuracy. Open route results are reported as executor-based SR / SPL. Overall Acc. is pooled across the six included accuracy tasks, excluding the two local motion tasks and open route narration. Overall values are provisional weighted estimates from the reported task accuracies and task counts; final pooled scores require verification of evaluation denominators and exact correct-answer counts.

Configuration	Route narration	Graph proximity	POI direction	POI actions	Visited POIs	Visited floors	Open route SR / SPL	Overall Acc.
Qwen3.8-Max								
Full-context	83.5	79.9	58.5	64.2	92.8	83.7	31.3 / 0.307	75.2
Text memory	64.10	79.80	39.40	49.00	86.40	55.80	11.6 / 0.105	60.3
Image–text memory	62.80	81.30	35.20	50.80	86.70	51.20	13.7 / 0.113	59.5

between landmark recall and the use of spatial relations for route planning, with further performance declines under the evaluated online-memory configurations. These findings highlight the importance of retaining spatial relations, alongside recognizable landmarks, in memory systems for subsequent reasoning and navigation.

AI USE STATEMENT

We used generative AI tools to assist with writing and language polishing, literature discovery, research ideation, and code development. Benchmark questions are instantiated from predefined templates, and reference answers are computed from spatial annotations and graph geometry rather than generated by an LLM. The authors are responsible for the accuracy of the manuscript, references, code, and reported results, including all AI-assisted content.

ETHICS STATEMENT

Before public release, we plan to screen the dataset for identifiable personal information, blur visible faces, remove embedded geolocation metadata, and use anonymous scene identifiers and scene-local coordinates. Any replacement of location or POI names will be applied consistently across released images, questions, answer options, reference answers, and annotations. If identifying text must be redacted from an image, the corresponding sample will be reviewed to ensure that the remaining visual evidence supports the task; samples that no longer do so will be revised or excluded. We will preserve the correspondence between visual landmarks and POI labels, and document any release-time changes that affect the evaluated data. We also plan to review visible location identifiers and map materials to determine whether further redaction or distribution restrictions are needed.

REPRODUCIBILITY STATEMENT

The main paper and appendix describe the benchmark construction, task definitions, evaluation protocols, scoring rules, memory configurations, and prompt templates. We plan to release the dataset and spatial annotations, subject to the privacy and distribution review described above, together with task-generation and evaluation code, model configurations, and per-question predictions and scores. The release will document model versions, input preprocessing, inference settings, and any differences between the evaluated data and the public release.

REFERENCES

Peter Anderson, Angel X. Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Shuran Song. On evaluation of embodied navigation agents, 2018a. URL https://arxiv.org/abs/1807.06757.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3674–3683, 2018b. URL https://arxiv.org/pdf/1711.07280.

Abrar Anwar, John Welsh, Yan Chang, Mayur Patel, and Siddhartha Srinivasa. Remembr: Building and reasoning over long-horizon spatio-temporal memory for robot navigation. *arXiv preprint arXiv:2409.13682*, 2024. URL https://arxiv.org/pdf/2409.13682. Also published at ICRA 2025.

Bohacek et al. Ovo-s-bench: A hierarchical benchmark for streaming spatial intelligence in multimodal llms, 2026. URL https://arxiv.org/html/2606.03890.

Chen et al. Mapgpt: Map-guided prompting with adaptive path planning for vision-and-language navigation, 2024. URL https://arxiv.org/html/2401.07314v3/.

Hongkai Chen et al. Spatiotemporal knowledge graphs as persistent scene memory for embodied question answering. *arXiv preprint arXiv:2510.01483*, 2025. URL https://arxiv.org/pdf/2510.01483.

Ding et al. Dynamic topology awareness: Breaking the granularity rigidity in vision-language navigation, 2026. URL https://arxiv.org/html/2601.21751.

Gordon et al. Multion: Benchmarking semantic map memory using multi-object navigation. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/pdf/2012.03912v1.pdf.

Yixu Hu, Zhicheng Zheng, and Dhruv Shah. Raven: Long-horizon reasoning navigation with a visuo-spatio-temporal memory. *arXiv preprint arXiv:2606.25206*, 2026. URL https://arxiv.org/pdf/2606.25206.

Huang et al. 3dllm-mem: Long-term spatial-temporal memory for embodied 3d large language model, 2025. URL https://arxiv.org/html/2505.22657v1.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Faiss: A library for efficient similarity searching and clustering, 2017. URL https://arxiv.org/abs/1702.08834.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *Computer Vision – ECCV 2020*, pp. 166–182, 2020. URL https://arxiv.org/pdf/2004.02857.

Lee et al. Egoexomem: Cross-view memory reasoning over synchronized egocentric and exocentric videos, 2026. URL https://arxiv.org/html/2605.18734v2.

Li et al. Dreamnav: A trajectory-based imaginative framework for zero-shot vision-and-language navigation, 2025a. URL https://arxiv.org/html/2509.11197v1.

Bohao Li et al. Capnav: Capability-conditioned navigation benchmark for multimodal large language models. *arXiv preprint arXiv:2511.06881*, 2025b. URL https://arxiv.org/pdf/2511.06881.

Liu et al. S³-bench: See, remember, explore: A benchmark and baselines for streaming spatial reasoning, 2026. URL https://arxiv.org/html/2603.23864v1.

Smith et al. Rem: Evaluating llm embodied spatial reasoning through multi-frame trajectories, 2025. URL https://arxiv.org/html/2512.00736v1.

Teney et al. Spamem: Benchmarking dynamic spatial reasoning via perception–memory integration in embodied environments, 2026. URL https://arxiv.org/html/2604.22409v3.

Wang et al. Boosting zero-shot vln via abstract obstacle map-based waypoint prediction with topograph-and-visitinfo-aware prompting, 2025. URL https://arxiv.org/html/2509.20499.

Wang et al. Topo-aerovln: Cognitive topological mapping for brain-inspired aerial vision-language navigation. OpenReview ICLR submission, 2026a. URL https://openreview.net/forum?id=IqbCMSrz0x.

Yixin Wang et al. Mind over space: Can multimodal large language models mentally navigate? *arXiv preprint arXiv:2603.21577*, 2026b. URL https://arxiv.org/pdf/2603.21577.

Zihan Xu et al. Explore with long-term memory: A benchmark and multimodal llm-based reinforcement learning framework for embodied exploration. *arXiv preprint arXiv:2601.10744*, 2026. URL https://arxiv.org/pdf/2601.10744.

Yang et al. Gst-bench: Can vlms develop global spatial awareness from video?, 2026. URL https://arxiv.org/html/2608.05747v1.

Yin et al. Egomonth: A month-level egocentric video benchmark for long-term spatiotemporal memory, 2026. URL https://arxiv.org/html/2608.13113v1.

Zhang et al. Msnav: Zero-shot vision-and-language navigation with dynamic memory and llm spatial reasoning, 2025. URL https://arxiv.org/html/2508.16654v1/.

Ruochen Zhang et al. Thinking in space: How multimodal large language models see, remember, and recall spaces. *arXiv preprint arXiv:2412.14171*, 2024. URL https://arxiv.org/pdf/2412.14171.

Hao Zheng et al. Spatialbench: Benchmarking multimodal large language models for spatial cognition. *arXiv preprint arXiv:2511.21471*, 2025. URL https://arxiv.org/pdf/2511.21471.

Zhou et al. Endowing embodied agents with spatial reasoning capabilities for vision-and-language navigation, 2025. URL https://arxiv.org/html/2504.08806.

Zhou et al. User-centric continual spatial intelligence reasoning in egocentric video streams, 2026. URL https://arxiv.org/html/2606.15200.

A Task Definitions and QA Generation

A.1 Notation and Observation Conventions

We represent each environment as a topological graph $G = (V, E)$, whose nodes carry scene-local coordinates, floor labels, and POI annotations, and whose edges denote traversable connections. A trajectory $\tau = (v_0, \ldots, v_T)$ is an ordered sequence of observation nodes; a trajectory group contains multiple trajectories from the same environment. The initial reference heading is randomly selected. At each non-initial node v_t, the reference heading follows the arrival direction $v_{t-1} \rightarrow v_t$, with directional views and egocentric answers defined relative to this heading. Models receive rendered observations with explicit trajectory boundaries and question-specific inputs. The full reference graph is used for answer generation and scoring. Memory information access is described in Appendix C.1.

A.2 Task Definitions

For graph proximity and route-related tasks, reference paths are computed on a task-specific graph containing connections traversed by the input trajectories together with visible, traversable connections that were not traversed. Shortest paths are computed within this graph.

Frame Displacement (`frame_displacement`). Given multi-view observations at two consecutive positions, the model selects the displacement between their observation positions from four candidate answers. The reference displacement is the Euclidean distance between the annotated positions.

Frame Direction (`frame_direction`). Given multi-view observations at two consecutive positions, the model selects the signed change in camera heading from four candidate answers. Positive angles indicate right turns, and negative angles indicate left turns.

Visited POI Recall (`visited_poi`). Given trajectory observations, the model determines whether a queried POI was passed. Positive queries sample POIs from the annotated visit sequence, while negative queries sample unvisited POIs from the same scene. The reference answer is a binary yes/no label. A POI counts as visited only when the trajectory passes through its corresponding annotated node; visibility from another node alone does not count as a visit.

Visited Floor Recall (`visited_floors`). The model identifies the set of floors visited across the provided trajectories. Reference answers are derived from the floor labels of visited nodes and evaluated without regard to order. The benchmark does not include half-floor labels.

POI Direction (`poi_direction`). Given trajectory observations and a current viewpoint, the model identifies the direction of a target POI relative to the current camera heading. Reference directions are computed from the annotated positions and quantized into eight angular bins. The model selects the correct direction from four candidate answers.

Immediate POI Action (`poi_action`). Given trajectory observations, a current viewpoint, and a target POI, the model selects the next movement direction along a shortest path to the target. Actions are expressed relative to the current camera heading.

Graph Proximity (`graph_proximity`). The model identifies which of four candidate POIs has the shortest walking distance to a reference POI. Reference distances are computed using shortest paths on the task-specific graph defined above, with Euclidean edge-length weights.

Route Navigation (`route_nav`). The model selects a natural-language description of a shortest route from the current viewpoint to a target POI from four candidate answers. Route descriptions specify landmark-based movements and turns, with the initial action defined relative to the current camera heading.

Open Route Navigation (`route_nav_open`). The model generates walking instructions from the current viewpoint to a target POI using a provided list of POI names. The task includes cases where a route uses a visible, traversable connection that was not traversed in the input trajectories. Generated instructions are evaluated through simulated execution using SR and SPL, as described in Appendix F.

QA Generation and Quality Control. Questions use predefined templates, and reference answers are computed programmatically from trajectory annotations and graph geometry without LLM-based generation. Heading-dependent answers use the reference heading of the selected viewpoint. For distance questions, distractors are generated by adding offsets to or subtracting offsets from the reference distance. For direction questions, distractors are selected from directions other than the reference answer. For multiple-choice route questions, distractors are generated by modifying turns, landmarks, or their order in the reference route instructions. All generated QA pairs undergo manual review. We exclude samples with missing required images, multiple valid answer options, or numerical options that are too close to distinguish reliably. Route options are checked for formatting consistency, including consecutive turn expressions, the placement of "turn around," and consistency of the target POI. Duplicate routes are filtered using start–end keys and instruction text. Questions are written in English, with POI names mapped through a scene-specific name table or a pinyin fallback.

B Model and Inference Settings

Model identities and evaluation scope. Table 4 lists the model names and their evaluation scope.

Table 4: Model registry. Main denotes Table 2; memory denotes Table 3, including its full-context reference.

Model	Evaluation
Qwen3.8-Max	Main, memory
Kimi K3	Main
Claude 5 Sonnet	Main
Qwen3-VL-2B	Main
Qwen3-VL-4B	Main
Qwen3-VL-8B	Main
Qwen3-VL-32B	Main

Observation inputs. Perspective views are rendered from panoramic observations with a horizontal field of view of $110°$. The first step of each trajectory supplies front, left, right, and back views; subsequent steps supply front, left, and right views. Full-context inputs contain all associated sampled observations in chronological order with explicit trajectory boundaries, followed by question-specific inputs. For tasks requiring a current viewpoint, both protocols receive the same current-view images. Online-memory construction uses the current views and a history window of three steps; historical evidence returned during answering depends on the memory configuration, as described in Appendix C.1.

Inference configuration. The recorded memory-QA configuration uses greedy decoding with temperature 0, seed 42, and a maximum of 4,096 output tokens. Memory observation calls use temperature 0.1 and a maximum of 4,096 output tokens, with up to 100 retries; retrieval budgets are listed in Table 6. These memory settings do not specify the full-context or executor decoding configuration.

B.1 Route Executor Configuration

The route-evaluation protocol uses Qwen3.8-Max as the shared executor across evaluated instruction-generating models. It follows generated and ground-truth reference instructions using the same execution prompt. Table 5 summarizes its observation inputs and episode settings.

Table 5: Route executor inputs and episode settings.

Setting	Value
Model name	Qwen3.8-Max (`qwen3.8-max`)
Current observations	Front, left, right, and back views; each a 1024×1024 JPEG image
Additional inputs	Generated route instructions, recent directional views, previous-step summary, and available movement directions
History-window length	Two preceding steps, with front, left, and right views at each step, together with the previous-step summary
Image resolution / preprocessing	1024×1024 views projected from panoramas with a $110°$ horizontal field of view; supplied as base64-encoded JPEGs without additional client-side resizing or normalization
Input size	Up to 10 images per step under the two-step history window; observed prompt size approximately 8,000–11,000 tokens per step
Episode budget	30 steps
Action space	Topology-permitted directions among `front`, `left`, `right`, and `back`, plus `stop`
Success condition	Issue `stop` at the target node within the episode budget (`arrival_mode=must_claim`)

The executor selects actions from current observations and the generated instructions; the simulator maps movement actions to adjacent nodes. The complete executor prompt is given in Appendix D.4, and SR/SPL definitions are given in Appendix F.

C MEMORY IMPLEMENTATION

C.1 TWO-STAGE PROTOCOL AND INFORMATION ACCESS

Memory-conditioned evaluation separates question-independent observation from question-conditioned answering. Each sample is replayed as a segmented observation sequence to construct one memory before its questions are revealed. The observation model receives the current multi-view images and a bounded history window; both memory configurations additionally receive the preceding segment summary. At answering time, the model receives the constructed memory, retrieval tools, and question-specific inputs. Current-view images, when required, are identical under full-context and memory evaluation.

Models and state across questions. The memory evaluation uses Qwen3.8-Max as the backbone. Captioning, step summaries, spatial-summary integration, and QA use the same VLM instance. Each question is answered independently using the same observation memory constructed for its sample, with a fresh conversation and QA-agent state.

C.2 TEXT MEMORY

Our text memory is inspired by ReMEmbR Anwar et al. (2024), particularly its use of textual descriptions of visual observations for semantic retrieval. We instantiate this approach for fixed trajectory observations and question-independent memory construction, using the hierarchical summaries and step records described below.

Construction. At each step, the model generates an environmental caption, a list of visible POIs, and an updated trajectory summary. The summary incorporates the preceding summary and the current observation to retain the route, landmark order, and current location. It is reset at each trajectory boundary. Step records are stored at the end of each trajectory, together with the final summary and a timeline of POI occurrences. After all trajectories have been observed, a further model call integrates the stored text into a spatial summary before any question is revealed. This produces three levels of memory: a spatial summary (L0), trajectory records (L1), and detailed step records (L2).

Retrieval and answering. Step captions are embedded for semantic retrieval, with trajectory and step identifiers preserving their temporal organization. The answering model initially receives the spatial summary and trajectory records. It can then query the memory store for relevant step records. Retrieval returns the top k distinct steps by embedding similarity, together with neighboring records from the same trajectory to preserve local route context. The returned captions and summaries supplement the initial context, and the model may request further evidence within its tool budget. Text memory retains no historical images; question-specific current-view images remain part of the task input.

Embeddings and retrieval indices. The configured embedding model is `qwen3-vl-embedding`, producing 2,560-dimensional vectors. Text and image embeddings are L2-normalized and searched with FAISS Johnson et al. (2017) `IndexFlatIP`, which performs exact inner-product search; for normalized vectors, inner-product similarity equals cosine similarity. Text and image–text memory retrieve evidence by embedding similarity without position-based lookup or annotated coordinates.

C.3 IMAGE–TEXT MEMORY

Our image–text memory is inspired by RAVEN Hu et al. (2026) in retaining visual evidence for subsequent reasoning. In our implementation, stored images augment the textual hierarchy used by text memory and are indexed by trajectory and observation step, with retrieval performed over both captions and images as described below.

Construction. Image–text memory uses the same caption and summary-update procedure as text memory, retaining the same three levels of textual information. It additionally stores the available

Table 6: Memory settings used by the reported implementation. Both configurations use the same listed settings.

Code parameter	Text	Image–text
`step_history_window` (h)	3	3
`stm_top_k` (k)	3	3
`window` (w, tool default)	1	1
`agent_max_tool_calls` (B)	5	5
`max_tool_turns`	4	4
`step_temperature`	0.1	0.1
`step_max_tokens`	4096	4096

directional images and their embeddings in a shared text–image embedding space. Each image is associated with its originating trajectory and observation step, allowing retrieved visual evidence to be interpreted alongside the corresponding caption and trajectory summary.

Retrieval and answering. The model receives the spatial summary and trajectory records and uses text queries to retrieve detailed evidence. Queries search both caption and image embeddings. Candidate entries are grouped by observation step and ranked by the highest similarity within each group, preventing multiple views of one position from occupying all retrieval slots. Selected steps return their text and available directional images; neighboring records provide additional temporal context. Thus, image–text memory differs from text memory in both its retrieval candidates and the evidence returned to the model. The answering procedure otherwise follows the same tool-based interaction.

C.4 Memory Configuration and Budgets

Table 6 reports the configured values, including defaults used when not overridden. The history window h is `step_history_window`; k is `stm_top_k`; w is the retrieval tool's `window`; and B is `agent_max_tool_calls`, a total tool-call cap rather than a tool-round cap. The separate `max_tool_turns` setting limits interaction rounds.

Observation calls allow up to 100 retries. QA uses greedy decoding with temperature 0, seed 42, and a maximum of 4,096 output tokens. Both memory configurations use one additional spatial-summary integration call per sample. Each configuration processes the same observation steps with one observation-model call per step, apart from retries. Text memory returns no historical images to QA, while image–text memory can return multiple images per retrieved step. All configurations may receive question-specific current views. Retrieval neighborhoods and accumulated tool results further affect context size. These settings therefore do not equalize total model calls, input tokens, image tokens, or computation across memory configurations.

D Experiment Prompts

This section presents the prompt templates used in our benchmark. Prompts and memory hyperparameters were refined during benchmark development and fixed before the final evaluation. Each template is fixed during evaluation. Placeholders enclosed by square brackets are dynamically filled for each test sample.

D.1 Full Context Experiment

D.1.1 Spatial Scene Understanding (Multiple Choice)

The prompt below is used for the spatial scene understanding multiple-choice subtask. The model is given multi-view trajectory observations and selects the correct answer from four options.

Spatial Scene Understanding Full Prompt Template

```
You are an indoor navigation scene-understanding assistant.
```

```
You receive one or more spatial roaming trajectories. The first
    observation step of each trajectory provides four directional
    images: front, left, right, back. Remaining observation steps
    provide front, left and right images only. All views are relative
    to the camera heading at that observation step.
Consecutive observations within the same trajectory are spatially
    adjacent along the walking path. You will also be given a question
     regarding these observations. Answer strictly based on the visual
     content from the provided images.
Requirements:
1. Examine all images carefully for store signs, signboards, POIs,
    area layouts and other visible visual cues.
2. Build spatial understanding solely from the given roaming-
    trajectory observations. You may only perform reasonable spatial
    reasoning derived from visual evidence in the images.
3. Do not fabricate any information that cannot be observed from the
    provided images.
--- History observations (total [T] steps, [S] segments) ---
Note: the images below are grouped by segment. Within the same
    segment, images with adjacent index are also spatially adjacent.
--- Traj 1 ---
--- index 0 ---     Front: [image]  Left: [image]  Right: [image]
    Back: [image]
--- index k ---     Front: [image]  Left: [image]  Right: [image]
--- Traj 2 ---
--- index 0 ---     Front: [image]  Left: [image]  Right: [image]
    Back: [image]
--- index k ---     Front: [image]  Left: [image]  Right: [image]
Your current position (the starting point) is shown below (left/front
    /right/back views). Use your current position to determine spatial
     orientation before answering.
Left: [image]  Front: [image]  Right: [image]  Back: [image]
Question: [question]
A. [option 1]
B. [option 2]
C. [option 3]
D. [option 4]
Answer only the option letter (e.g. A, B, C, D), no extra text.
```

D.1.2 Route Navigation (Open-Ended Generation)

This prompt is adopted for open-ended route navigation. The model outputs comma-separated walking instructions following the predefined format rules.

Full Open-Ended Route Navigation Prompt Template

```
You are an indoor navigation scene-understanding assistant.
You receive one or more spatial roaming trajectories. The first
    observation step of each trajectory provides four directional
    images: front, left, right, back. Remaining observation steps
    provide front, left and right images only. All views are relative
    to the camera heading at that observation step.
Consecutive observations within the same trajectory are spatially
    adjacent along the walking path. You will also be given a
    navigation target; based on these observations, output walking
    instructions starting from your current position.
Requirements:
1. Examine all images carefully for store signs, signboards, POIs,
    area layouts and other visible visual cues.
```

```
2. Build spatial understanding solely from the given roaming-
    trajectory observations. You may only perform reasonable spatial
    reasoning derived from visual evidence in the images.
3. Do not fabricate any information that cannot be observed from the
    provided images.
--- History observations (total [T] steps, [S] segments) ---
Note: the images below are grouped by segment. Within the same
    segment, images with adjacent index are also spatially adjacent.
--- Traj 1 ---
--- index 0 ---     Front: [image]  Left: [image]  Right: [image]
    Back: [image]
--- index k ---     Front: [image]  Left: [image]  Right: [image]
--- Traj 2 ---
--- index 0 ---     Front: [image]  Left: [image]  Right: [image]
    Back: [image]
--- index k ---     Front: [image]  Left: [image]  Right: [image]
Your current position (the starting point) is shown below (left/front
    /right/back views). Use your current position to determine spatial
     orientation before answering.
Left: [image]  Front: [image]  Right: [image]  Back: [image]
Question: [question]
Available stores: [store 1], [store 2], ...
Output only a single walking instruction, separated by commas, with
    no prefix or trailing explanation.
Format: {action} from current view, go straight to {landmark}, {
    action}, go straight to {landmark}, ..., go straight to {target}
Allowed action words: go straight, turn left, turn right, turn around
      write every action (including go straight);
Important rules:
1. The first move is decided by the current view: the front view is
    the direction you are facing. Target straight ahead  go straight
    first; on the left  turn left first; on the right  turn right
    first; directly behind  turn around first. Do not write "turn
    around" or "turn left" without basis;
2. Every real turn must be written out: at any junction that requires
     a turn on the route, you must write turn left / turn right / turn
     around; consecutive "go straight to A, go straight to B" that
    gloss over a turning section is prohibited;
3. Only move along landmarks that are visible in the images and
    appear in the "available stores" list; for uncertain directions,
    refer to the latest corridor/signboard you saw;
4. If a turning junction has no store name, refer to it relatively as
     "the intersection after {previous store}" (previous store = the
    one you just passed);
5. The last step must end with "go straight to {target store}".
Example 1 (go straight first + mid turn left): go straight from
    current view, go straight to [landmark], turn left, go straight to
     [target]
Example 2 (turn left first + mid turn right): turn left from current
    view, go straight to [landmark], turn right, go straight to [
    target]
Example 3 (turn around first): turn around from current view, go
    straight to [landmark], turn left, go straight to [target]
```

D.1.3 ROUTE NAVIGATION (MULTIPLE CHOICE)

This prompt is for the multiple-choice variant of route navigation. The model selects the correct walking instruction from four candidate options.

Full MCQ Route Navigation Prompt

```
You are an indoor navigation scene-understanding assistant.
You receive one or more spatial roaming trajectories. The first
    observation step of each trajectory provides four directional
    images: front, left, right, back. Remaining observation steps
    provide front, left and right images only. All views are relative
    to the camera heading at that observation step.
Consecutive observations within the same trajectory are spatially
    adjacent along the walking path. You will also be given a
    navigation target; based on these observations, output walking
    instructions starting from your current position.
Requirements:
1. Examine all images carefully for store signs, signboards, POIs,
    area layouts and other visible visual cues.
2. Build spatial understanding solely from the given roaming-
    trajectory observations. You may only perform reasonable spatial
    reasoning derived from visual evidence in the images.
3. Do not fabricate any information that cannot be observed from the
    provided images.
--- History observations (total [T] steps, [S] segments) ---
Note: the images below are grouped by segment. Within the same
    segment, images with adjacent index are also spatially adjacent.
--- Traj 1 ---
--- index 0 ---     Front: [image]  Left: [image]  Right: [image]
    Back: [image]
--- index k ---     Front: [image]  Left: [image]  Right: [image]
--- Traj 2 ---
--- index 0 ---     Front: [image]  Left: [image]  Right: [image]
    Back: [image]
--- index k ---     Front: [image]  Left: [image]  Right: [image]
Your current position (the starting point) is shown below (left/front
    /right/back views). Use your current position to determine spatial
     orientation before answering.
Left: [image]  Front: [image]  Right: [image]  Back: [image]
Question: [question]
Available stores: [store 1], [store 2], ...
Output only a single walking instruction, separated by commas, with
    no prefix or trailing explanation.
Format: {action} from current view, go straight to {landmark}, {
    action}, go straight to {landmark}, ..., go straight to {target}
Allowed action words: go straight, turn left, turn right, turn around
      write every action (including go straight);
Important rules:
1. The first move is decided by the current view: the front view is
    the direction you are facing. Target straight ahead  go straight
    first; on the left  turn left first; on the right  turn right
    first; directly behind  turn around first. Do not write "turn
    around" or "turn left" without basis;
2. Every real turn must be written out: at any junction that requires
     a turn on the route, you must write turn left / turn right / turn
     around; consecutive "go straight to A, go straight to B" that
    gloss over a turning section is prohibited;
3. Only move along landmarks that are visible in the images and
    appear in the "available stores" list; for uncertain directions,
    refer to the latest corridor/signboard you saw;
4. If a turning junction has no store name, refer to it relatively as
     "the intersection after {previous store}" (previous store = the
    one you just passed);
5. The last step must end with "go straight to {target store}".
Example 1 (go straight first + mid turn left): go straight from
    current view, go straight to [landmark], turn left, go straight to
     [target]
```

```
Example 2 (turn left first + mid turn right): turn left from current
    view, go straight to [landmark], turn right, go straight to [
    target]
Example 3 (turn around first): turn around from current view, go
    straight to [landmark], turn left, go straight to [target]
Question: [question]
A. [option walk-instruction 1]
B. [option walk-instruction 2]
C. [option walk-instruction 3]
D. [option walk-instruction 4]
Answer only the option letter (e.g. A, B, C, D), no extra text.
```

D.2 Local Motion Estimation (Multiple Choice)

This prompt is used for local motion estimation subtask. The model infers camera movement from consecutive ordered multi-view frames and answers the multiple-choice question.

Full Local Motion Estimation MCQ Prompt

```
You are an indoor motion perception assistant. You are given a short
    sequence of ordered observation frames captured during continuous
    walking. Each frame contains front/left/right views depending on
    what is actually provided, together with a question about this
    movement. Answer the question based only on the content of these
    observation frames.
Requirements:
1. Each pair of consecutive observation steps corresponds to one
    movement. Infer the movement distance and turn by comparing the
    position and scale changes of shops, corridors and landmarks
    between adjacent observation steps.
2. Left turns, right turns and turn-arounds are defined relative to
    the camera's forward direction (front view).
3. When estimating distance, make reasonable qualitative judgments
    using visible scene references such as door width, store signs and
     floor tiles. Do not invent precise numeric values.
4. Do not invent information about the full trajectory beyond the
    provided observation steps.
--- History observations (total [F] frames) ---
--- index 0 ---    Front: [image]  Left: [image]  Right: [image]
    Back: [image]
--- index 1 ---    Front: [image]  Left: [image]  Right: [image]
...
--- index [F]-1 ---  Front: [image]  Left: [image]  Right: [image]
Question: [question]
A. [option 1]
B. [option 2]
C. [option 3]
D. [option 4]
Answer only the option letter (e.g. A, B, C, D), no extra text.
```

D.2.1 Template Variable Definitions

All bracketed placeholders are replaced with sample-specific content during evaluation:

- `[T]`: total number of observation steps across all trajectories.
- `[S]`: total number of independent trajectory segments.
- `[F]`: total number of observation frames for local motion estimation.
- `[image]`: placeholder for multi-view image input.
- `[question]`: spatial reasoning / motion estimation question for the sample.

- `[store 1], [store 2], ...`: white list of valid POI landmarks for route navigation.
- `[option 1]--[option 4]`: four candidate multiple-choice options.

D.3 TRACE PROMPT

D.3.1 TEXT AND TEXT+IMAGE

Trace Step Observation Prompt (Text / Text+Image)

```
You are an indoor navigation scene-understanding assistant. Below are
     the **images in various directions** of the current node (may
    include multiple views front/left/right/back).
If there are historical step images, they are arranged before the
    current step in chronological order, with the current step's
    images last; combine them with history, note the continuity
    between consecutive views, the route already walked, and whether
    you have returned to a previously visited place.

Integrate all these images and describe the environment around the
    current node in **one coherent passage (caption)**  do not write
    per-direction separately.
Include: names and signs of nearby stores, signboards, notices, area
    features, floor information, landmarks and other concrete visual
    content; be specific and searchable  this text will later be used
    for memory retrieval (remembr style).

**Notes**:
- caption must be one coherent passage covering the key information
    from all views, kept to 80~200 characters
- include the content of every view as much as possible; do not miss
    any informative image
- do not use per-direction enumeration such as "on the left I saw..."
     / "ahead..."; weave what you see into one overall description
- summary is the **cumulative summary of the whole walk within this
    segment from the start to the current step**: build on the
    previous step's summary (if any), add this step's new information,
     and condense the full route, order and landmarks; new content
    supersedes old, do not only write this step
- reference style: first overview the whole route, then land on the
    current position, e.g. "walked from Gate 7  atrium  escalator bank
    , now near the escalators on floor 5; passed: Wu Zhuang, Gu Ming,
    Uniqlo"
- keep it concise with route/landmark phrases; do not restate per-
    step image details; this summary is injected into the next step as
     context to keep accumulating

Output as JSON:
{
  "caption": "whole description of the environment around the current
     node",
  "seen_pois": ["clearly visible stores/place names; empty array if
    none"],
  "summary": "cumulative summary of the whole segment's route +
    current position"
}

This is step [t] of the navigation trajectory (total [T] steps).
    Observe **all** directional images around the current node and
    output one coherent passage describing the surroundings.
```

```
Accumulated trajectory summary of all previous steps (from the
    previous step): [prev_summary]   (from step 1 on)

--- History frames (recent [W] steps, oldest to newest) ---
Step [t-W]:  Front: [image]  Left: [image]  Right: [image]
...
--- Current step ---
Front: [image]  Left: [image]  Right: [image]  Back: [image]   (step
    0 only)
Output your observations in JSON format.
```

D.4 ROUTE EXECUTION

Route Execution Prompt

```
You are a visual navigation model. Each turn you receive:
1. Four directional images of the current node (front / left / right
    / back);
2. Images from up to two previous steps (History step, front / left /
     right);
3. The summary from the previous turn and this turn's available
    directions.

Use the instruction, the current directional images, the history
    images and the summary
to judge your location and progress, and pick the next direction to
    move; if you have
reached the target position, output stop instead of picking a
    direction.

Directions map one-to-one to the images (front = straight ahead, left
     = left side,
right = right side, back = behind you). Do not guess node ids, and do
     not pick a
direction that was not provided.

Each turn, build on the previous summary and your current observation
     to summarize in
one or two sentences where you are, which key landmarks you see, and
    your next plan,
keeping a coherent progress log across turns.

Output protocol (plain text, no JSON or markdown):
Reply with exactly two lines, the first starting with "summary", the
    second with "decision".

summary: <In one or two sentences, based on the previous summary, the
     current images and the history images, summarize your location,
    the key landmarks you see and your next step>
decision: <front | left | right | back | stop>
- decision must be one of the available directions for this turn; use
     stop when you are sure you have reached the target.

Instruction: turn around from current view, go straight to Futi
    Entrance, turn right, go straight to witoon, go straight to FUGA
Previous summary: Having turned around from the original view (PANE
    store and escalators now behind me), I'm facing the opposite
    corridor with the DESCENTE store and the atrium railing on my left
    ; this is the straight path toward Futi Entrance.
Available directions: front (A0), back (A1)
```

```
--- History step -2 (previous): ---  Front: [image]  Left: [image]
    Right: [image]
--- History step -1 (previous): ---  Front: [image]  Left: [image]
    Right: [image]
...
View front: [image]  View left: [image]  View right: [image]  View
    back: [image]
```

E Error Analysis

We inspect model predictions alongside the input observations and annotated topology to characterize route-planning failures. Our analysis focuses on whether predicted instructions are consistent with the starting viewpoint, landmark directions, and navigable connections. These cases illustrate observable failure patterns rather than uniquely identifying their internal causes.

Localization and Viewpoint Association. We observe a route-prediction failure consistent with incorrect association of the current viewpoint with the environment. To reach Keyu, the reference route requires turning around, proceeding to the first intersection, and turning left. The model also begins with a turn-around, but then describes a different landmark sequence through Xianglala, Keming Cafe, and Xian Tan. The discrepancy therefore extends beyond the initial turn: the subsequent instructions do not match the reference route from the given viewpoint. This case suggests that producing plausible landmark-based instructions does not necessarily imply correct grounding of the route in the current observation.

F Evaluation Metrics

Answer Accuracy. For all tasks except `route_nav_open`, predictions are compared with ground-truth answers using task-specific answer-matching rules, without a model-based judge. For multiple-choice questions, the selected option letter is extracted from the response and compared with the reference option; accompanying explanations are ignored when a unique selected option can be parsed. For POI recall, yes/no answers are matched after case and punctuation normalization. Visited-floor answers are evaluated by exact set matching, ignoring order. Empty responses, unparseable answers, and multiple-option answers are scored as incorrect. We report per-task accuracy:

$$\mathrm{Acc}_t = \frac{1}{N_t} \sum_{i=1}^{N_t} \mathbb{I}[\mathrm{match}_t(\hat{y}_i, y_i)] \,,$$

where N_t is the number of questions in task t, and match_t denotes its answer-matching rule.

Let $\mathcal{T}_{\mathrm{acc}}$ denote all evaluated tasks except `route_nav_open`. Overall accuracy is pooled across these tasks:

$$\mathrm{Acc}_{\mathrm{overall}} = \frac{\sum_{t \in \mathcal{T}_{\mathrm{acc}}} N_t \mathrm{Acc}_t}{\sum_{t \in \mathcal{T}_{\mathrm{acc}}} N_t}.$$

Execution-Based Route Evaluation. For `route_nav_open`, models generate natural-language walking instructions. The Qwen3.8-Max navigation executor follows these instructions step by step in a simulator grounded in the environment topology and egocentric observations used to construct the benchmark. At each step, the executor receives the generated instructions, current directional views, recent observations, the previous-step summary, and available movement directions. It selects a movement direction or issues a stop action; the simulator maps each movement to an adjacent node. Movement choices are restricted to the available directions. Each episode is limited to 30 steps and succeeds only if the executor issues a stop action at the target node within this budget. Stopping elsewhere or exhausting the budget without a successful stop counts as failure. No task requires passing through a designated connection; routes may use visible, traversable connections that were not traversed in the input trajectories. Execution-based scores reflect both the generated instructions and the executor's ability to follow them.

We report success rate (SR) and success weighted by path length (SPL) Anderson et al. (2018a), measuring goal-reaching success and navigation efficiency, respectively:

$$\mathrm{SR} = \frac{1}{N}\sum_{i=1}^{N} S_i, \qquad \mathrm{SPL} = \frac{1}{N}\sum_{i=1}^{N} S_i \frac{\ell_i}{\max(p_i, \ell_i)},$$

where N is the number of open-ended route questions, S_i is a binary execution-success indicator, p_i is the executed path length computed by summing the lengths of all traversed edges, and ℓ_i is the shortest-path distance from the starting position to the target on the task-specific graph. The shortest-path reference has no mandatory-waypoint or mandatory-connection constraint. SR and SPL are reported separately and are excluded from overall answer accuracy.

G EXAMPLES

This section presents examples of nine task types in TrajsCogMap-Bench, covering frame-level motion understanding, spatial reasoning, action and direction reasoning, and route navigation. Each example illustrates the task input, question, and reference answer. For visualization, we display selected observations and include explanatory annotations that are excluded from the model inputs.

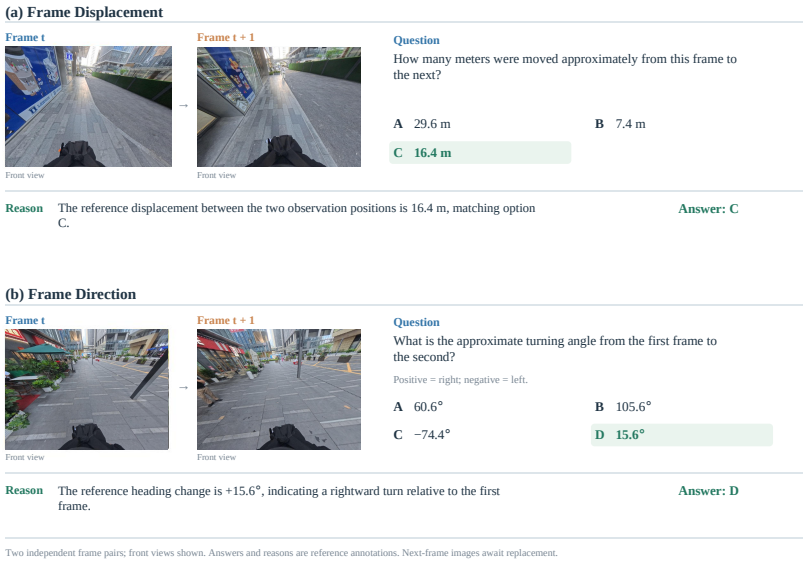


Figure 4: Examples of frame-level motion questions in TrajsCogMap-Bench. (a) Frame Displacement evaluates distance estimation between consecutive observation positions. (b) Frame Direction evaluates signed heading-change estimation, with positive angles indicating right turns. The examples use separate observation pairs; only front views are shown for compactness. Highlighted answers and explanations are reference annotations and are not provided to the models.

H LIMITATIONS AND FUTURE WORK

Environment coverage and visual cues. The benchmark covers 33 non-overlapping locations in POI-rich indoor environments. This scope does not establish generalization to outdoor navigation, sparsely landmarked spaces, or changing environments. Store signs and POI names provide useful anchors, but also make performance sensitive to text recognition, language familiarity, and landmark visibility. Broader scene coverage and tasks with fewer textual cues would help distinguish these factors from spatial reasoning.

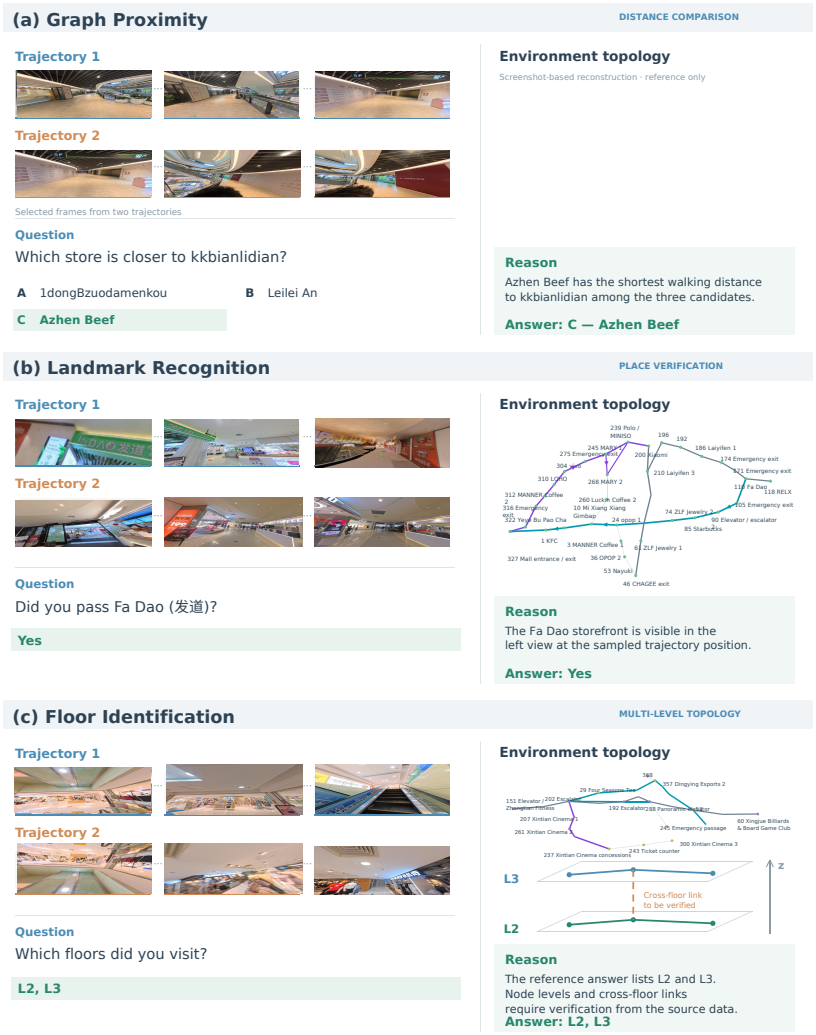


Figure 5: Examples of spatial reasoning tasks in TrajsCogMap-Bench. (a) Graph Proximity requires selecting, from four candidate POIs, the one with the shortest walking distance to a target POI. (b) Visited POI Recall requires determining whether the observed trajectories passed through the specified POI's annotated node; visibility from another node alone does not count as a visit. (c) Floor Identification requires identifying the floors visited across the trajectories. Each task provides two egocentric observation trajectories. Only selected trajectory frames are shown for compactness. The topology diagrams, reference answers, and explanations are included for illustration and are not provided to the evaluated models. The layered topology in (c) schematically illustrates the vertical floor dimension.

Open-loop evaluation. Fixed observation trajectories enable comparisons using the same visual evidence, but do not evaluate active exploration, observation selection, or recovery through new observations. Discrete topology-based route execution also abstracts away continuous robot control and physical obstacles. Extending the benchmark to interactive settings would test whether the spatial knowledge measured here supports navigation under these additional demands.

Execution-based scoring. Open-ended route scores depend on both the generated instructions and the executor's ability to ground and follow them. Correct instructions can fail during execution, so SR and SPL do not isolate route-planning quality. The reference-instruction execution results in

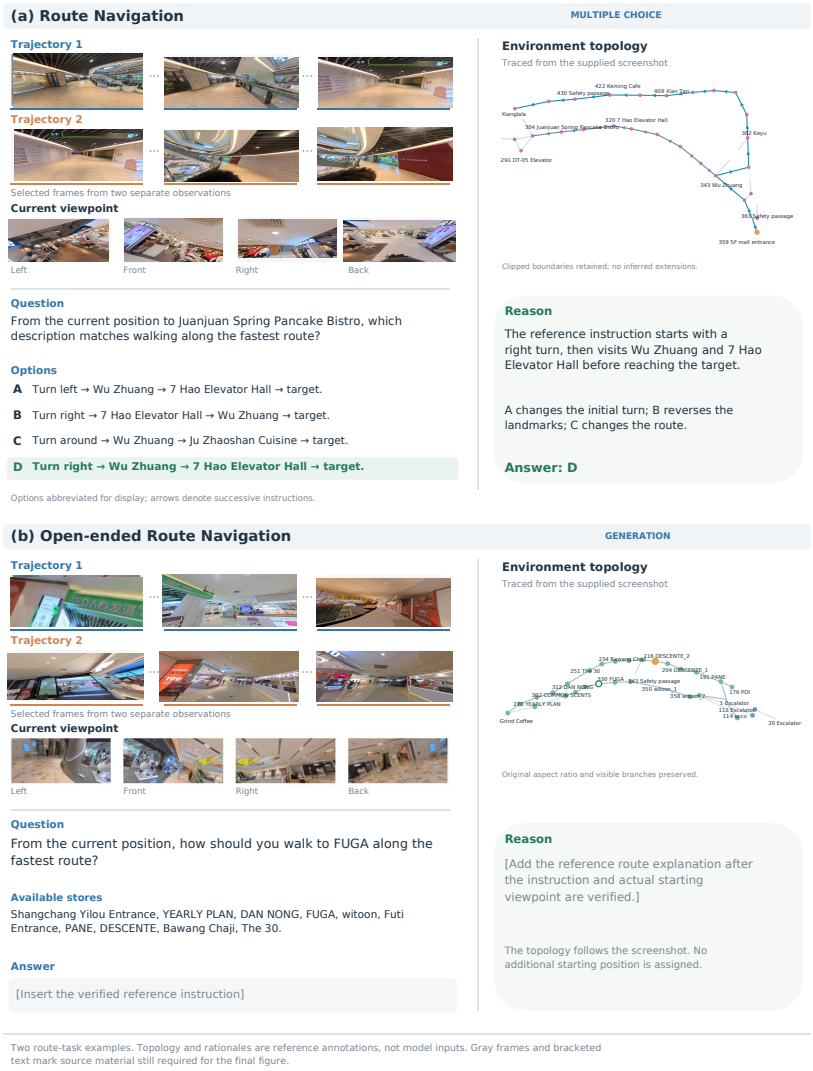


Figure 6: Examples of route navigation tasks in TrajsCogMap-Bench. (a) Route Navigation requires selecting the correct route description from four candidates. (b) Open-ended Route Navigation requires generating landmark-based walking instructions to the target POI. Both tasks provide two egocentric observation trajectories and the current viewpoint. Only selected trajectory frames are shown for compactness. The topology diagrams, reference answers, and explanations are included for illustration and are not provided to the evaluated models.

Table 2 provide a calibration point for the shared executor, but do not remove its influence on model comparisons. Comparisons across executors would further characterize this source of variation.

Interpretation of task and memory performance. Correct spatial answers do not by themselves establish that a model has formed an internally consistent cognitive map. Likewise, task errors do not uniquely distinguish perception, association, memory construction, retrieval, and reasoning failures. The multi-trajectory setting evaluates spatial knowledge use after multiple traversals; individual questions may rely on subsets of those observations, so aggregate accuracy does not establish that every answer integrates all trajectories. The memory configurations also differ in retrieval candidates and access to stored images, so their performance differences cannot be attributed solely to memory structure. Future evaluations could more directly test cross-question consistency, dependence on complementary trajectory evidence, and memory performance under matched resource budgets.