ROOM TO REMEMBER: KNOWLEDGE-GRAPH MEMORY FOR LONG-HORIZON EMBODIED QUESTION ANSWERING

Anonymous authors
Paper under double-blind review

ABSTRACT

An assistant asked about a room it has already left cannot answer from its current image. We study how a vision-language model should read a retained record of those observations. Our benchmark in simulated ProcTHOR houses pairs recorded trajectories with questions about object locations, room visits, temporal order, and accumulated observations. Predicted scene graphs record objects, rooms, relations, and sighting times; we train answerers to read them as text, learned graph features, or both. Training and evaluation use blank answer images to separate memory access from answer-time vision. On 800 questions from 200 test episodes, the textual graph-memory representation improves MomaGraph-R1 accuracy by 6.62 percentage points over a no-memory control across two training seeds, an effect concentrated in route-history questions (+18.9 points) with little change on the rest (-0.4). Learned graph features alone show no detectable improvement over no memory on either backbone at either seed. These comparisons favor text within the tested architectures and training schedules. Blind probes fall below chance on the families carrying the gain and well above it on the rest, so aggregate accuracy reflects both retained evidence and question priors. The results support training a text reader as a baseline for episodic memory.

1 INTRODUCTION

An assistant that has crossed a house may be asked whether it saw a fridge before a houseplant, or which objects were visible in an earlier room. Its current image cannot answer these questions. It needs a record of past observations and a way to use that record when answering.

Recent embodied systems retain observations in episodic memories and scene graphs, records of objects, rooms, and their relations (Saxena et al., 2025; Ginting et al., 2025; Parashar et al., 2026). Their environments, backbones, and training procedures differ, making the memory interface difficult to compare across papers. Room to Remember studies a narrower choice: how a trained vision-language model should read a record built from the same source observations.

Text can list the observed objects, relations, and sighting times using the pretrained model's language interface. A graph encoder instead maps the record to feature vectors that the answerer must learn to use. Surveys identify this distinction for language models reading graphs (Jin et al., 2024; Yu et al., 2026). We ask: *Should a pretrained vision-language model read an episodic scene-graph record as rendered text, learned graph features, or both?*

Room to Remember tests supervised question answering over recorded trajectories through simulated ProcTHOR (Deitke et al., 2022) houses. Predicted scene graphs accumulate within each episode; four configurations expose no memory, rendered text, learned graph features, or both, with shared observations and training budgets. Every configuration answers with a blank image, separating record use from answer-time vision. The comparison concerns reading an available history; autonomous exploration is evaluated separately.

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.

Across two backbones, trained text memory achieves the highest aggregate accuracy among the tested interfaces. Its gains concentrate in questions about earlier sightings and room visits, without increasing monotonically with route length. Learned graph access shows no detectable aggregate gain over no memory. Family analyses, image-free probes, and label audits identify where the gains occur and how question priors and reference labels affect their interpretation.

Our contributions are threefold. First, Room to Remember provides a controlled testbed for this comparison: four memory interfaces sharing source observations, backbone, and training budget, evaluated with blank answer images to isolate record use from visual grounding at answer time. Second, we show that a trained text interface outperforms both a no-memory control and learned graph features on two backbones, with the gain concentrated in questions about route history. Third, image-free probes, family-level sensitivity checks, and paired error analysis characterize the sources and scope of the gains. A warmup diagnostic tests recovery from an observed graph-only collapse under a revised gate schedule.

2 Related work

Embodied and episodic question answering. EmbodiedQA studies navigation to answer questions (Das et al., 2018); OpenEQA distinguishes episodic recall from active evidence acquisition (Majumdar et al., 2024). GraphEQA combines hierarchical scene graphs and images for active EQA (Saxena et al., 2025), Mind Palace links temporally indexed world instances across episodes (Ginting et al., 2025), and VL-KnG queries a persistent tour graph (Mdfaa et al., 2026). Exploration-aware evaluation and efficient exploration separately examine evidence acquisition (Jiang et al., 2025; Ren et al., 2024). We study supervised answering over recorded trajectories, holding answer-time visual access fixed; Appendix Table 43 compares benchmark designs.

Memory interfaces. MomaGraph provides our primary scene-graph backbone (Ju et al., 2026). MrSteve organizes what, where, and when information (Park et al., 2025); EPM learns textual state updates (Parashar et al., 2026), whereas our renderer is rule-based and the answerer learns to consume its output. JanusVLN and MemoryVLA study navigation and manipulation memories (Zeng et al., 2026; Shi et al., 2026); GraphPad and VL-MemKnG combine graphs with text (Ali et al., 2025; Lukina et al., 2026). ReMEmbR builds spatio-temporal video memory for navigation questions (Anwar et al., 2025), evaluating retrieval and reasoning end to end. We instead hold the record fixed and vary its interface to one fine-tuned answerer. Other systems construct multi-view or open-vocabulary records (Yang et al., 2025b; Gu et al., 2024), organize spatial and temporal observations (Nguyen et al., 2026), select task-relevant or surprising events (Chang et al., 2026; Gorlo et al., 2026), or verify plans (Wang et al., 2026). Their varied interfaces motivate our comparison; we do not reproduce or rank these systems.

Evaluation and shortcut controls. LongMemEval, REMem, and Memory-T1 examine conversational and temporal memory (Wu et al., 2025; Shu et al., 2026; Du et al., 2026); World-in-World evaluates closed-loop behavior (Zhang et al., 2026). Ref-Adv tests visual shortcut reliance (Dong et al., 2026), and Lost in Serialization shows sensitivity to how a graph is written (Herbst et al., 2025). We use family-level analysis, image-free probes, and separate evidence-acquisition measurements to interpret memory accuracy. Appendix P gives the extended background and comparisons.

3 Trajectory-based question answering

An episode consists of observations and actions $\tau = (I_0, a_0, \ldots, I_{T_\tau - 1}, a_{T_\tau - 1})$ collected in a simulated house. A question q has a finite option set $\mathcal{C}_q$ and a reference answer y_q. We study prediction after the traversal, $\hat{y}_q = f_\theta(q, \mathcal{C}_q, M, \bar{I})$, where M is the retained observation record and $\bar{I}$ is a fixed blank image. Earlier images inform memory construction. They are not provided as answer-time visual evidence in the supervised comparison.

Episode generation. We record ProcTHOR-10k trajectories (Deitke et al., 2022) in AI2-THOR 5.0.0 (Kolve et al., 2017), using six discrete actions and 256×256 RGB frames. Traversal precedes question construction, so questions cannot influence the route. Strata S1–S4 have zero, one, two,

or at least three door crossings; they measure route length, not validated difficulty. Navigation outcomes document traversal but are not evaluated here (Appendix C).

Question writing. Symbolic templates read the simulator's per-frame visible-object log to produce questions, reference answers, and justifications over a sampled strip of up to six frames. Memory instead uses predicted scene graphs over all retained frames. Each question has two to four shuffled options. Table 1 defines the eight families; Appendix C gives their operational rules and schema checks. The generation prompt omits the navigation target object and room.

Table 1: **The eight question families.** Counts are test questions; abbreviations are used in results. Route-history is a descriptive analysis grouping.

Family	Abbrev.	n	What the reference answer encodes
Route-history: sightings and room visits along the route			
Order	Order	73	Which of two object types was first seen earlier
Presence-change	Pres-chg	102	An object seen earlier and absent at the final sampled frame
Rooms-visited	Visited	116	How many distinct rooms the route entered
Set: object presence, location, and counts			
Persistence	Persist	131	In how many distinct rooms an object type was seen
Room Identification	RoomID	80	The room label associated with sightings of a type
Existence	Exist	113	Whether a type was observed in a named room
Counting	Count	65	Distinct instances of a type observed in one room
Global-count	Glob-cnt	120	Distinct instances of a category seen anywhere on the route

Validation checks. Stage A filters target mentions and drops contradictions identified by a symbolic answer checker. Stage B applies blind and options-only probes to measure leakage in the released split, rather than filter it. Four families are probe-sensitive, covering 397 of the 800 test questions; they remain in the evaluation (Appendix K).

Evaluation population. Training, validation, and test contain 5,000, 500, and 200 episodes. Test episodes have four questions each: 800 questions, with 200 per route stratum. Splits are house-disjoint, checked through object identity rather than split-local `house_idx` values (Appendix H). Question-weighted chance is 33.0%; a constant answer letter is a separate baseline. Frame totals and option counts are in Appendix C.

Intended memory demand and observed validity. For exploratory analysis, Order, Presence-change, and Rooms-visited form the *route-history* group (291 questions); the other five families contain 509. This is descriptive: counting visited rooms need not use visit order. Family names and the geometric `memory_required` flag describe intended evidence use; image-free controls assess how much can be answered from priors (Appendix C).

Human reference and label audit. Four annotators answered 50 questions (25 each from Presence-change and Global-count), with reference answers hidden and every episode frame visible. The families separate: all four scored above the 50.0% binary chance level on Presence-change (56.5–66.7%, majority vote 70.0%), though no single interval excludes chance; on Global-count they span 17.4–44.4% around 28.0% chance and their majority vote of 31.2% is itself at chance. Agreement is near chance and *falls* as readers are added, from $\kappa = 0.180$ with two annotators to $\kappa = 0.114$ with four. The reference rules use the sampled strip, while wording refers to the episode or house. Because annotators saw the whole episode, this is neither a label-error rate nor a human ceiling (Appendix G.3; Table 9).

Known limitations of the artifact. The sampled strip indices are not persisted in the released records, preventing reconstruction of the exact evidence behind a label. We retain both audited families and report sensitivity analyses excluding them in Section 5.2. Presence-change illustrates the scope problem: an object can leave view and reappear before the final sampled frame, satisfying the wording but not the terminal-absence rule. Global-count wording can refer to the house while its reference covers only the sampled strip. These differences matter when interpreting agreement with

the reference as successful recall. Appendix G.5 gives the reference explanations and distinguishes scope mismatch from a measured label-error rate.

4 EPISODIC MEMORY AND ITS INTERFACES

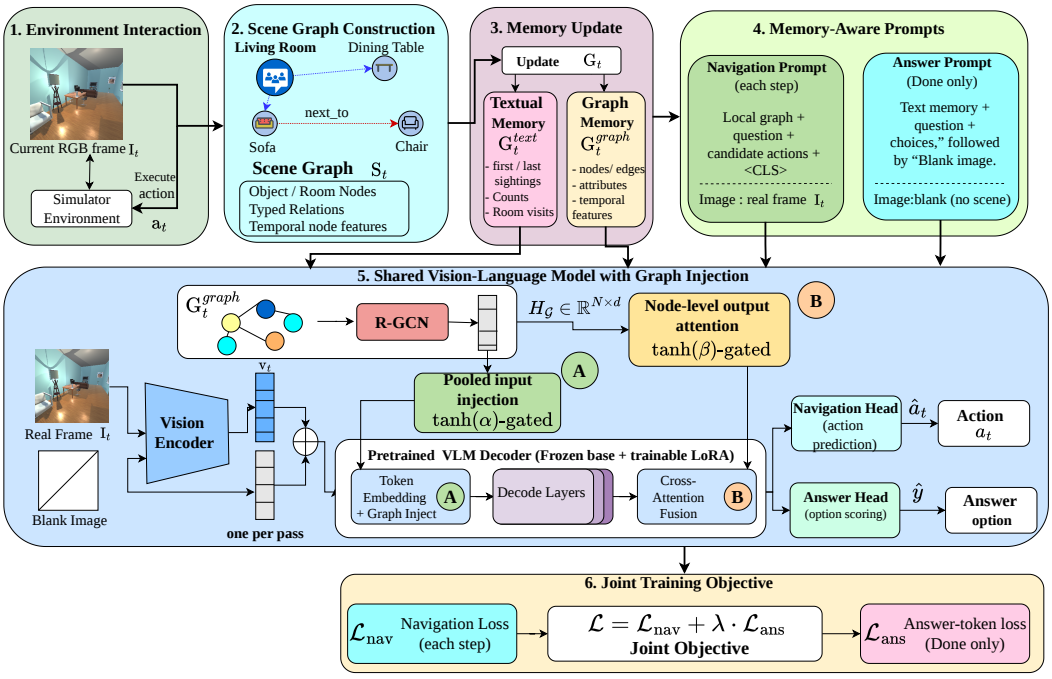


Figure 1: **From observations to memory-conditioned answers.** Configuration D uses text plus pooled input injection and node-level output attention. Navigation sees real images; answering sees a blank image. Both heads share the decoder. Appendix D gives dimensions and settings.

4.1 CONSTRUCTING A RECORD OF OBSERVATIONS

For observation I_t, a prompted vision-language model predicts $S_t = P_\phi(I_t, q_1)$ using the first episode question as task context. The parser distinguishes the observed room from the task-implied target room. Conditioning extraction on q_1 is a property of the pipeline rather than of any single question: the same record answers all m questions of the episode, so memory cannot be tailored per question, though it may be biased toward the content q_1 names. For T retained observations, define

$$M = (S_0, \ldots, S_{T-1}), \qquad G = \mathcal{B}(M) = (V, \mathcal{E}, X), \qquad \mathbf{r} = \mathcal{R}(M). \tag{1}$$

The builder $\mathcal{B}$ accumulates typed nodes V, node features X, and deduplicated typed relations $\mathcal{E}$. It links objects to rooms, retains predicted spatial relations, and adds room adjacency from consecutive visits. The renderer $\mathcal{R}$ independently summarizes the ordered scene-graph records as text. At step t, G_t is built from the retained prefix M_t; the final record excludes the Done image. Memory resets between episodes. Canonical node keys merge types without tracking physical instances.

Text readout. For each room and object type, text reports first and last sightings and maximum simultaneous count. Frequency prioritizes entries under the budget of 12 types per room and 24 relations. Consecutive equal room labels become visits with frame spans, preserving return visits. A record such as `Bed[0-1]` gives endpoints, not continuous visibility.

Graph features. The graph has 104 object types, four room types, and 12 relation types. Its 85-dimensional node features include a learned name embedding and six temporal features: normalized first and last sightings, span, frequency, and initial- and final-frame presence. Relations retain accumulated spatial and room structure (Appendix D).

4.2 READING MEMORY THROUGH TEXT AND GRAPH FEATURES

The text path inserts $\mathcal{R}(M)$ alongside the question and full options. The graph path uses two relational graph-convolution layers (Schlichtkrull et al., 2018), mapping $85 \to 128 \to 256$ dimensions, and projects node states to $H_G \in \mathbb{R}^{|V|\times d}$. It supplies residuals before and after the shared decoder:

$$\tilde{E} = E + \tanh(\boldsymbol{\alpha}) \odot \mathrm{LN}(\mathrm{MLP}(\mathrm{mean}_v(H_G))), \tag{2}$$

$$\tilde{H} = H + \big(\tanh(\beta) + b(s)\big)\mathrm{MHA}(\mathrm{LN}_q(H), \mathrm{LN}_{kv}(H_G), \mathrm{LN}_{kv}(H_G)). \tag{3}$$

Here E contains token embeddings and $H \in \mathbb{R}^{L\times d}$ the decoder states after combining text and image tokens. The pooled residual broadcasts across embeddings; eight-head output cross-attention accesses individual nodes. LN denotes layer normalization. Base weights are frozen, with LoRA updates on W_q, W_k, W_v, W_o. The per-channel input gate initializes to 0.02 and the scalar output gate to zero. During training, $b(s)$ rises from 0 to 0.2 over w steps, then falls to zero over $2w$; inference sets it to zero. The offset affects only the output gate. Recovered MomaGraph-R1 arm C uses $w = 2000$; other runs use $w = 400$ (Appendix E).

Configurations and comparison. Let $(b_{\text{text}}, b_{\text{graph}})$ select text and graph access. A/B/C/D use $(0,0)$, $(1,0)$, $(0,1)$, and $(1,1)$, respectively (Appendix Table 6). Disabling graph access bypasses both residuals. B–A measures the trained text configuration's utility; D–B measures the effect of adding graph access to it. The source observations are shared, while filters and budgets differ between the two representations. C drops architectural graph nodes and D retains them, so C–D also changes graph content and we do not report it as a text ablation. The graph configurations use a gate warmup schedule whose length we set per configuration; the values, and the sensitivity of arm C to that choice, are given in Appendix E.

Appendix N walks through a question, its retained record, and each configuration's answer.

4.3 JOINT SUPERVISION AND ANSWER SCORING

The navigation prompt contains the real image, local scene-graph text, question context, six candidate actions, and a classification token. A normalized linear head reads this token and predicts MoveAhead, RotateLeft, RotateRight, LookUp, LookDown, or Done. The answer prompt uses the blank image, enabled memory text, question, and two to four options. For m episode questions, answer supervision applies only on Done samples:

$$\mathcal{L}_t = -\log p_\theta(a_t \mid I_t, S_t, q_1, G_t) + \mathbf{1}\{a_t = \text{Done}\}\frac{\lambda}{m}\sum_{j=1}^{m}\mathcal{L}_{\text{ans},j}. \tag{4}$$

Graph conditioning is bypassed in A/B. The language-model head supplies answer logits; training uses full-vocabulary cross-entropy, while evaluation compares only the supplied answer-letter logits. Full option texts remain in the prompt. Navigation is an auxiliary loss shared across arms and is not evaluated. Scene-graph extraction remains a separate prompted step.

Runs use $\lambda = 1$, eightfold Done oversampling, and LoRA (Hu et al., 2022) rank 16. Each configuration trains for 6,000 steps on one NVIDIA H200 (140 GiB), with the final checkpoint evaluated and no validation-based selection. Appendices D–F give dimensions, hyperparameters, and compute.

5 EXPERIMENTS

We evaluate three questions. Does trained graph-derived text improve answer accuracy? Which question families benefit? Does learned graph access add value, either alone or alongside text? All supervised comparisons use the same 800 questions and blank answer images. MomaGraph-R1 (Ju et al., 2026) and Gemma-3-4B (Gemma Team, 2025) each use two seeds in all four configurations.

Statistical protocol and scope. For binary correctness c^X_{esj} of configuration X, episode e, matched seed s, and question j, define

$$d^{X-Y}_{ej} = \frac{1}{|S|}\sum_{s\in S}(c^X_{esj} - c^Y_{esj}), \qquad \widehat{\Delta}^{X-Y}_{Q} = \frac{\sum_{e,j}\mathbf{1}\{(e,j)\in Q\}d^{X-Y}_{ej}}{\sum_{e,j}\mathbf{1}\{(e,j)\in Q\}}. \tag{5}$$

We bootstrap 200 episode indices with replacement 20,000 times, keeping each episode's four questions together and recomputing the ratio for each subset Q. The 2.5th and 97.5th percentiles give 95% intervals. Seeds are paired before averaging, so each question contributes once to the resampling unit. Intervals measure episode-sampling variation conditional on these checkpoints, excluding variation across new seeds and dependence from repeated houses. Family and route analyses are post hoc and unadjusted for multiplicity. Earlier representation development inspected a test split; its question overlap with the present evaluation is unresolved, so this study is exploratory. Appendix H gives the estimator; Tables 13 and 14 list the runs and contrasts.The reduced-set audit in Table 11 is the one exception and resamples questions rather than episodes.

5.1 Trained graph-derived text improves answer accuracy

Text memory improves both evaluated backbones (Table 2). MomaGraph-R1 reaches 71.44% with text against 64.81% without memory, a gain of 6.62 percentage points [3.94, 9.38]. Both seeds improve, by 7.25 and 6.00 points. Gemma-3-4B reaches 66.69% against a no-memory control of 63.69%, a gain of 3.00 points [0.69, 5.31]. Both configurations have two seeds, so this is a matched contrast, but the effect is less than half the size of MomaGraph-R1's and we describe it as a consistent direction on a second backbone rather than a replication of the magnitude.

B–A compares trained configurations under a fixed answer-image condition. Memory changes both the prompt and training inputs, so isolating correct episode content would require a matched content intervention. Without fine-tuning, the same memory text moves MomaGraph-R1 from 48.38% to 41.63%, a loss of 6.75 points, while with fine-tuning the identical input is worth a 6.6-point gain. The sign reversal supports training the answerer to use the record. It does not isolate the value of correct episode content, and the archived untuned invocation is incomplete (Appendix H).

5.2 The gain concentrates in route-history families

The aggregate improvement comes primarily from Order, Presence-change, and Rooms-visited (Table 2). On this route-history group, MomaGraph gains 18.90 points [12.81, 24.92] and Gemma gains 8.08 [3.27, 13.20]; the other five families change little (-0.39 and $+0.10$, both intervals spanning zero). The differences between group effects are 19.29 [12.93, 25.62] and 7.98 [2.62, 13.51] points.

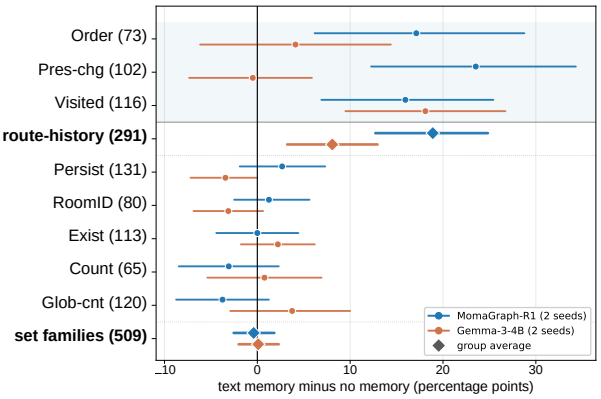


Figure 2: **Text-memory effects by question family.** Text memory minus no memory, both backbones, two matched seeds each, against their own seed-matched no-memory control. Bars are paired episode-bootstrap 95% intervals; parentheses give question counts. The shaded families form the descriptive route-history group; diamonds give question-weighted group averages. Both backbones clear zero on the route-history group and neither does on the set group.

The composition reconstructs the aggregate. For MomaGraph-R1, 291 route-history questions at $+18.90$ and 509 remaining questions at -0.39 average to 6.63 points over 800, against the 6.6 reported above. The aggregate thus combines a larger effect on 36% of the benchmark with near-zero effects on the remaining families.

The individual gains differ by backbone (Figure 2). MomaGraph gains 23.53 points on Presence-change; Gemma gains nothing there (-0.5), and has its largest gain on Rooms-visited at 18.1. Counting and Global-count show no consistent benefit. Removing Rooms-visited leaves a 20.86-point improvement on Order and Presence-change for MomaGraph, compared with 1.43 points for Gemma, whose interval includes zero.

We also check whether the MomaGraph result depends on Presence-change, whose reference label rule uses absence at the final observation. Excluding that family leaves a 4.15-point text-over-

Table 2: **Accuracy by question family on the 800-question test set.** Families and their abbreviations are defined in Table 1; route-history is a descriptive grouping of sightings and room visits. Zero-shot rows answer from the question's referenced evidence frames, fine-tuned rows from a blank image, so the two regimes do not share an evidence protocol and are *not* one ranked leaderboard: **bold** and underline mark best and second-best per column *within each block*, and each backbone's four configurations are ranked against each other rather than across backbones. Group averages are question-weighted over the family sizes in the header; chance is question-weighted and varies by family. **S** is seeds averaged. Against its own seed-matched no-memory control, text memory gains +18.9 [12.9, 25.0] on route-history families and −0.4 [−2.5, 1.8] on set families for MomaGraph-R1, and +8.1 [3.3, 12.9] and +0.1 [−2.0, 2.3] for Gemma-3-4B; every graph-memory contrast includes zero on both groups and both backbones.

		Route-history families				Set families						
Method	S	Order (73)	Pres-chg (102)	Visited (116)	**AVG** (291)	Persist (131)	RoomID (80)	Exist (113)	Count (65)	Glob-cnt (120)	**AVG** (509)	**All** (800)
Chance (question-weighted)	–	50.0	50.0	28.9	41.6	30.0	28.2	25.1	28.6	28.6	28.1	33.0
• *Language only, no images*												
Qwen3-4B (Qwen Team, 2025a)	–	39.7	51.0	30.2	39.9	38.2	70.0	61.1	26.2	32.5	45.4	43.4
• *Zero-shot vision-language models, referenced evidence frames*												
Qwen2.5-VL-32B (Team, 2025)	–	78.1	52.0	**72.4**	**66.7**	**71.0**	66.2	69.9	49.2	**31.7**	**58.0**	**61.1**
Qwen3-VL-4B (Qwen Team, 2025b)	–	74.0	40.2	67.2	59.5	68.7	66.2	**73.5**	**50.8**	30.0	**58.0**	58.5
Qwen3-VL-8B (Qwen Team, 2025b)	–	**80.8**	49.0	60.3	61.5	**71.0**	65.0	67.3	43.1	30.0	56.0	58.0
Qwen2.5-VL-7B (Team, 2025)	–	61.6	40.2	69.0	57.0	61.8	**68.8**	68.1	40.0	29.2	53.8	55.0
Gemma-3-4B-IT (Gemma Team, 2025)	–	41.1	**52.9**	55.2	50.9	60.3	55.0	64.6	36.9	**31.7**	50.7	50.8
LLaVA-1.6-7B (Liu et al., 2024)	–	63.0	49.0	23.3	42.3	67.2	38.8	68.1	41.5	23.3	49.3	46.8
Idefics3-8B (Laurençon et al., 2024)	–	56.2	37.3	43.1	44.3	55.7	45.0	64.6	41.5	26.7	47.3	46.2
MomaGraph-R1† (Ju et al., 2026)	–	42.5	42.2	28.4	36.8	54.2	51.2	69.9	35.4	26.7	48.3	44.1
• *Socratic pipeline: caption, then answer from the captions*												
LLaVA-1.6-7B (Liu et al., 2024) → Qwen3-4B (Qwen Team, 2025a)	–	41.1	43.1	34.5	39.2	48.9	60.0	48.7	23.1	24.2	41.5	40.6
• *Fine-tuned, blank answer image (ours) — Gemma-3-4B-IT* (Gemma Team, 2025)												
No memory	2	52.1	**61.8**	52.6	55.7	80.9	81.9	85.4	61.5	32.9	68.3	63.7
+ graph memory	2	52.1	60.8	50.0	54.3	**82.4**	82.5	86.3	**63.1**	33.8	**69.4**	63.9
+ text memory	2	**56.2**	61.3	**70.7**	**63.7**	77.5	78.8	**87.6**	62.3	**36.7**	68.4	**66.7**
+ text and graph	2	55.5	60.8	69.0	62.7	77.5	**83.1**	**87.6**	60.8	34.2	68.3	66.2
• *Fine-tuned, blank answer image (ours) — MomaGraph-R1* (Ju et al., 2026)												
No memory	2	54.1	58.3	52.2	54.8	80.5	83.1	86.3	**66.2**	**38.8**	**70.5**	64.8
+ graph memory	2	54.1	57.8	48.7	53.3	81.7	83.1	**88.5**	64.6	34.2	70.0	63.9
+ text memory	2	**71.2**	**81.9**	68.1	**73.7**	**83.2**	**84.4**	86.3	63.1	35.0	70.1	**71.4**
+ text and graph	2	62.3	66.7	**68.5**	66.3	79.0	**84.4**	87.6	65.4	36.2	69.9	68.6

†MomaGraph-R1's zero-shot row answers from its *last* referenced frame only, a limitation of its single-image QA adapter, and is not comparable to the other zero-shot rows. Parse failures count as incorrect throughout; LLaVA-1.6-7B and the Socratic pipeline produced a readable answer on 733 and 777 of 800 questions respectively, every other row on at least 798. Bold and underline rank within blocks: each backbone's four configurations are ranked against each other, so the marks read down a backbone rather than across the two.

no-memory gain [1.48, 6.91]. Excluding both families whose reference labels we audit in Appendix G, Presence-change and Global-count, 222 of the 800 questions leaves gains of 5.80 points on MomaGraph-R1 and 3.46 on Gemma-3-4B over the remaining 578, both intervals excluding zero. Appendix I gives family contrasts; Table 11 reports the reduced-set check. The audit uses question resampling. The gain does not grow monotonically with route length: it is largest in S1 and smaller in S4. Re-rendering the record at nine budgets spans 2.00 points; deleting every spatial relation costs 0.75 points [−2.50, +1.00] (Appendix Figure 6).

5.3 Text is the strongest evaluated memory interface

Text alone achieves the highest aggregate accuracy on both backbones (Table 2). Its advantage over text plus learned graph access is 2.81 points [0.56, 5.06] for MomaGraph and 0.44 points [−1.00, 1.81] for Gemma. Excluding Presence-change reduces the MomaGraph difference to 1.00 point [−1.08, 3.13]. Text is the strongest point estimate, while the uncertainty depends on the backbone and question set.

Graph memory alone shows no detectable improvement over no memory. Against seed-matched no-memory controls, the graph-only configuration scores -0.12 and -1.62 points on MomaGraph-R1 and $+0.00$ and $+0.38$ on Gemma-3-4B. All four intervals include zero. Eight further intervals after splitting by route stratum also span zero. These are inconclusive contrasts, not equivalence tests; they do not rule out smaller effects.

The graph-only result persists after repairing an observed optimization failure. One graph-only run did fail to train, converging on a near-constant answer letter. We diagnosed the gate schedule, changed that single setting, and the fusion gate opened by more than an order of magnitude; the recovered run reproduces at a second seed and still does not beat its control. The collapse was specific to that configuration: the combined text-and-graph configuration converges under the original schedule on both backbones, as does Gemma's graph-only configuration. Appendices E and J.1 give the schedules and the per-arm gate values; Figures 8 and 9 show the recovery. This intervention tests recovery from the observed collapse; the combined arm's gate settles well below the recovered run's and its schedule was not searched.

At seed 0, near-equal aggregate accuracy masks different answers: MomaGraph's graph-only configuration loses 33 control-correct answers and recovers 32; Gemma loses and recovers 25 each. The graph perturbs which questions are answered correctly without changing how many. Appendix J reports these transitions.

Adding graph access to the text configuration reduces the route-history gain from 18.90 to 11.51 points on MomaGraph-R1 and from 8.08 to 7.04 on Gemma, with little change on the remaining families. Per question, the graph costs MomaGraph 42 and 47 route-history answers across its two seeds and returns 21 and 25; on Gemma it costs 27 and 22 and returns 18 and 25. Both graph pathways are active, so these changes cannot identify pooling as the cause. Appendix J also distinguishes changes in correctness from changes in the selected answer letter.

5.4 What memory repairs, and what remains

Text memory repairs 107 and 100 no-memory errors across the two MomaGraph seeds, while introducing 49 and 52 regressions (Table 28). Seventy-two repairs recur on the same questions in both seeds. Gemma repairs 85 and 74 errors across its two seeds, with 57 and 54 regressions. The residual analysis covers all four text-memory runs (Table 36). Global-count comprises 15% of questions but accounts for 34.2% and 34.1% of MomaGraph's text-memory errors and 27.0% and 30.1% of Gemma's. Of 142 questions missed by all four text-memory runs, 55 are Global-count. This concentration motivates instance-aware counting as a targeted extension, without identifying the source of each error. Global-count is also one of the two families whose reference labels we audit in Appendix G, where no individual accuracy interval excludes chance on it. The question scope and reference rules therefore complicate attribution of these errors. Appendix L adds family breakdowns, seed overlap, and execution-level cases.

5.5 Validity checks and evidence acquisition

In this separate zero-shot comparison, referenced oracle frames provide the question's annotated evidence images; oracle replay instead obtains images by executing the recorded action sequence.

On Qwen3-VL-4B (Qwen Team, 2025b), an image-free probe supplied with the question, options, and target-room label scores 45.4%; an options-only probe scores 39.0% (Table 24). Both exceed the 33.0% question-weighted chance level. The target-room label supplies contextual metadata. Chance also varies by family: Order is binary, with exploitable option cues despite its temporal subject matter. These probes justify retaining image-free baselines when evaluating memory. Table 2 gives the zero-shot panel and Table 26 reports answer coverage. Differences across that panel combine evidence access with protocol-dependent parsing effects (Appendix K), so identifying image use on individual questions requires matched controls.

Evidence access and autonomous acquisition also produce different outcomes. Referenced oracle frames, which supply the question's annotated evidence images, score 58.5%, and oracle replay, which obtains images by executing the recorded action sequence, scores 57.4%, while closed-loop exploration scores 45.5% with no recorded navigation successes. Oracle replay supplies 198 usable trajectories from 200 attempts. These protocols evaluate evidence availability separately from super-

vised memory access. In the MomaGraph closed-loop run, QA is invoked only after Done: 128/800 questions receive parsed answers, with 54/128 correct (42.2%) and 54/800 correct overal(6.75%). The other 672 records follow stuck termination without answer invocation. This separates answer availability from formatting and correctness. The supervised gains concern reading retained evidence; autonomous exploration is evaluated by these separate runs.

6 Discussion

What the interface exposes.

The text renderer exposes room membership, visit spans, and first/last sightings in the pretrained model's language space. The graph reader receives the same attributes with typed connectivity and must learn to use them. The results favor training a text reader as a baseline. Pooling and node-level attention operate together, leaving their separate effects unresolved; the warmup diagnostic distinguishes optimization recovery from an accuracy gain.

These results concern recorded trajectories in simulation, with two seeds on each of two backbones. Family analyses are exploratory; autonomous evidence acquisition is evaluated separately. Type-based memory and the audited reference rules limit instance-counting conclusions (Appendix Q). The record builder could process real RGB observations without privileged simulator state, while real trajectories would require independently annotated references. Testing transfer requires evaluating perception quality, label scope, and whether the interface ordering and effect sizes persist.

What the comparison isolates. Source observations and answer-time images are shared. Text and graph differ in filters, budgets, gate schedules, and decoder interaction, so the comparison concerns these complete reader configurations.

What the record can distinguish. Global-count errors motivate distinguishing maximum simultaneous counts from distinct physical instances. A chair seen early and another seen later can leave the same maximum as repeated views of one chair; summing frame counts can count one object twice. First/last sightings omit intermediate visibility changes. Instance identity and event histories could address these limits, but attributing individual errors requires labels aligned with the requested counting operation (Appendix Q).

What would resolve the remaining ambiguity. Further controls would match nodes and temporal attributes across interfaces, ablate each graph pathway, and sweep the gate schedule across both graph arms. Three others require no retraining: replacing correct memory with a matched irrelevant record, extracting the record without conditioning on the episode's first question, and building it from the simulator's visibility log rather than predicted scene graphs. Together these would separate episode content from format, optimization, and extraction noise. Removing spatial relations causes little measured change; removal or permutation of timestamps, room grouping, and object lists remains untested (Figure 6). Appendix Q discusses these requirements, and Appendix O separates completed experiments from proposed controls.

7 Conclusion

Room to Remember compares how pretrained vision-language models read one episodic scene-graph record. Trained text improves accuracy by 6.62 points on MomaGraph-R1 and 3.00 on Gemma-3-4B, with gains concentrated in route-history families. Learned graph features alone change which questions are answered correctly but add no observed aggregate advantage across two backbones and two seeds, including after recovery of a collapsed graph-only run. Smaller graph effects remain compatible with the reported intervals. The study supports training a text reader as a baseline, then evaluating richer interfaces through matched answer changes and the information each representation preserves. We release the benchmark, episode records, and per-question predictions, including the two families whose reference rules our audit finds mis-scoped.

REPRODUCIBILITY STATEMENT

The appendix specifies training settings, evaluation conditions, statistical estimators, and reference prompts. Aggregate results and paired analyses can be recomputed from the exported per-question correctness records. Appendix D distinguishes the reference implementation from the logged runtime protocol. End-to-end reproduction additionally uses the deployed trainer, full episodes, house manifest, checkpoints, and predicted scene-graph cache. The authors retain the full episodes and deployed training source.

Code: https://anonymous.4open.science/r/ICLR_NAVIGRAPH-9557/README.md

ETHICS STATEMENT

The study uses simulated household scenes to examine algorithmic memory behavior. Image-free probes characterize language and environment priors in generated questions. Use in real homes would require consent and controls on retaining images and derived memory.

AI USE STATEMENT

Generative AI assisted literature discovery, manuscript drafting and revision, architecture-diagram formatting, code inspection, analysis-script development, and interpretation of experimental records. The benchmark pipeline uses a vision-language model for question construction alongside fallback templates and symbolic checks; model-generated scene graphs supply inputs to memory. Reported analyses are grounded in code, logs, and prediction records. The authors are responsible for this submission.

REFERENCES

Muhammad Qasim Ali, Saeejith Nair, Alexander Wong, Yuchen Cui, and Yuhao Chen. GraphPad: Inference-time 3D scene graph updates for embodied question answering. *arXiv preprint arXiv:2506.01174*, 2025.

Abrar Anwar, John Welsh, Joydeep Biswas, Soha Pouya, and Yan Chang. ReMEmbR: Building and reasoning over long-horizon spatio-temporal memory for robot navigation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2838–2845. IEEE, 2025.

Yue Chang, Rufeng Chen, Yifan Tian, Dazhi Huang, Zhaofan Zhang, Yi Chen, Wenze Zhang, Li Chen, Hui Xiong, and Sihong Xie. Just-in-time scene graph growth: Combating perceptual saturation in long-horizon robotics. *arXiv preprint arXiv:2607.13245*, 2026.

Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–10. IEEE, 2018.

Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Jordi Salvador, Kiana Ehsani, Winson Han, Eric Kolve, Ali Farhadi, Aniruddha Kembhavi, and Roozbeh Mottaghi. ProcTHOR: Large-scale embodied AI using procedural generation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 5982–5994, 2022.

Qihua Dong, Kuo Yang, Lin Ju, Handong Zhao, Yitian Zhang, Yizhou Wang, Huimin Zeng, Jianglin Lu, and Yun Fu. Ref-Adv: Exploring MLLM Visual Reasoning in Referring Expression Tasks. In *International Conference on Learning Representations*, volume 2026, pp. 105393–105409, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/ac37a79cc303ca982703f14caa033d8b-Paper-Conference.pdf.

Yiming Du, Baojun Wang, Yifan Xiang, Zhaowei Wang, Wenyu Huang, Boyang Xue, Bin Liang, Xingshan Zeng, Fei Mi, Haoli Bai, Lifeng Shang, Jeff Z. Pan, Yuxin Jiang, and Kam-Fai Wong.

Memory-T1: Reinforcement Learning for Temporal Reasoning in Multi-session Agents. In *International Conference on Learning Representations*, volume 2026, pp. 58583–58613, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/6010a57158b77359f5f531d9af69181c-Paper-Conference.pdf.

Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. Talk like a graph: Encoding graphs for large language models. In *International conference on learning representations*, volume 2024, pp. 43909–43934, 2024.

Gemma Team. Gemma 3 Technical Report, 2025. URL https://arxiv.org/abs/2503.19786.

Muhammad Fadhil Ginting, Dong-Ki Kim, Xiangyun Meng, Andrzej Marek Reinke, Bandi Jai Krishna, Navid Kayhani, Oriana Peltzer, David Fan, Amirreza Shaban, Sung-Kyun Kim, Mykel Kochenderfer, Ali-akbar Agha-mohammadi, and Shayegan Omidshafiei. Enter the mind palace: Reasoning and planning for long-term active embodied question answering. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 5072–5106. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/ginting25a.html.

Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. IQA: Visual question answering in interactive environments. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4089–4098. IEEE, 2018.

Nicolas Gorlo, Derek K Wise, Alberto Speranzon, and Luca Carlone. Worth remembering: Surprise-gated robot episodic memory. *arXiv preprint arXiv:2606.03787*, 2026.

Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. ConceptGraphs: Open-vocabulary 3D scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5021–5028. IEEE, 2024.

Daniel Herbst, Lea Karbevska, Divyanshu Kumar, Akanksha Ahuja, Fatemeh Gholamzadeh Nasrabadi, and Fabrizio Frasca. Lost in Serialization: Invariance and Generalization of LLM Graph Reasoners, 2025. URL https://arxiv.org/abs/2511.10234.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Kaixuan Jiang, Yang Liu, Weixing Chen, Jingzhou Luo, Ziliang Chen, Ling Pan, Guanbin Li, and Liang Lin. Beyond the destination: A novel benchmark for exploration-aware embodied question answering. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9091–9101. IEEE, 2025.

Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. Large language models on graphs: A comprehensive survey. *IEEE Transactions on Knowledge and Data Engineering*, 36 (12):8622–8642, 2024.

Yuanchen Ju, Yongyuan Liang, Yen-Jen Wang, Nandiraju Gireesh, Yuanliang Ju, Seungjae Lee, Qiao Gu, Elvis Hsieh, Furong Huang, and Koushil Sreenath. MomaGraph: State-aware unified scene graphs with vision-language model for embodied task planning. In *International Conference on Learning Representations*, volume 2026, pp. 96361–96386, 2026.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. AI2-THOR: An interactive 3D environment for visual AI. *arXiv preprint arXiv:1712.05474*, 2017.

Hugo Laurençon, Andrés Marafioti, Victor Sanh, and Léo Tronchon. Building and Better Understanding Vision-Language Models: Insights and Future Directions, 2024. URL https://arxiv.org/abs/2408.12637.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Wensi Ai, Benjamin Martinez, Hang Yin, Michael Lingelbach, Minjune Hwang, Ayano Hiranaka, Sujay Garlanka, Arman Aydin, Sharon Lee, Jiankai Sun, Mona Anvari, Manasi Sharma, Dhruva Bansal, Samuel Hunter, Kyu-Young Kim, Alan Lou, Caleb R. Matthews, Ivan Villa-Renteria, Jerry Huayang Tang, Claire Tang, Fei Xia, Yunzhu Li, Silvio Savarese, Hyowon Gweon, C. Karen Liu, Jiajun Wu, and Li Fei-Fei. BEHAVIOR-1K: A Human-Centered, Embodied AI Benchmark with 1,000 Everyday Activities and Realistic Simulation, 2024. URL https://arxiv.org/abs/2403.09227.

Wenlong Liang, Rui Zhou, Yang Ma, Bing Zhang, Songlin Li, Yijia Liao, and Ping Kuang. Large model empowered embodied AI: A survey on decision-making and embodied learning. *arXiv preprint arXiv:2508.10399*, 2025.

Yixin Lin, Jan Humplik, Sandy H. Huang, Leonard Hasenclever, Francesco Romano, Stefano Saliceti, Daniel Zheng, Jose Enrique Chen, Catarina Barros, Adrian Collister, Matt Young, Adil Dostmohamed, Ben Moran, Ken Caluwaerts, Marissa Giustina, Joss Moore, Kieran Connell, Francesco Nori, Nicolas Heess, Steven Bohez, and Arunkumar Byravan. Proc4Gem: Foundation Models for Physical Agency through Procedural Generation, 2025. URL https://arxiv.org/abs/2503.08593.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. LLaVA-NeXT: Improved Reasoning, OCR, and World Knowledge, January 2024. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

Svetlana Lukina, Mohamad Al Mdfaa, Gloria Haro, Sergey Zagoruyko, and Gonzalo Ferrer. VL-MemKnG: Hybrid memory with a spatio-temporal knowledge graph for question answering over long egocentric navigation trajectories. *arXiv preprint arXiv:2606.17183*, 2026.

Weijian Ma, Shizhao Sun, Tianyu Yu, Ruiyu Wang, Tat-Seng Chua, and Jiang Bian. Thinking with blueprints: Assisting vision-language models in spatial reasoning via structured object representation. *arXiv preprint arXiv:2601.01984*, 2026.

Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, et al. OpenEQA: Embodied question answering in the era of foundation models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16488–16498. IEEE, 2024.

Mohamad Al Mdfaa, Svetlana Lukina, Timur Akhtyamov, Arthur Nigmatzyanov, Dmitrii Nalberskii, Sergey Zagoruyko, and Gonzalo Ferrer. Spatiotemporal knowledge graphs as persistent scene memory for embodied question answering, 2026. URL https://arxiv.org/abs/2510.01483.

Phuoc Nguyen, Francesco Verdoja, and Ville Kyrki. Event-grounding graph: Unified spatio-temporal scene graph from robotic observations. *IEEE Robotics and Automation Letters*, 2026.

Priyam Parashar, Jacob Krantz, Matthew Chang, Kavit Shah, Xavier Puig, and Roozbeh Mottaghi. Planning with an embodied learnable memory. In *International Conference on Learning Representations*, volume 2026, pp. 142387–142400, 2026.

Junyeong Park, Junmo Cho, and Sungjin Ahn. MrSteve: Instruction-following agents in Minecraft with what-where-when memory. In *International Conference on Learning Representations*, volume 2025, pp. 16922–16960, 2025.

Bryan Perozzi, Bahare Fatemi, Dustin Zelle, Anton Tsitsulin, Mehran Kazemi, Rami Al-Rfou, and Jonathan Halcrow. Let your graph do the talking: Encoding structured data for llms. *arXiv preprint arXiv:2402.05862*, 2024.

Qwen Team. Qwen3 Technical Report, 2025a. URL https://arxiv.org/abs/2505.09388.

Qwen Team. Qwen3-VL Technical Report, 2025b. URL https://arxiv.org/abs/2511.21631.

Allen Z. Ren, Jaden Clark, Anushri Dixit, Masha Itkina, Anirudha Majumdar, and Dorsa Sadigh. Explore until confident: Efficient exploration for embodied question answering. In *Proceedings of Robotics: Science and Systems*, 2024.

Dennis Rotondi, Francesco Argenziano, Sebastian Koch, Nathan Hughes, Martin Büchner, Johanna Wald, Lukas Rosenberger Schmid, Daniele Nardi, Abhinav Valada, Liam Paull, Federico Tombari, Luca Carlone, and Kai O. Arras. 3D Scene Graphs: Open Challenges and Future Directions, 2026. URL https://arxiv.org/abs/2606.19383.

Saumya Saxena, Blake Buchanan, Chris Paxton, Peiqi Liu, Bingqing Chen, Narunas Vaskevicius, Luigi Palmieri, Jonathan Francis, and Oliver Kroemer. GraphEQA: Using 3D semantic scene graphs for real-time embodied question answering. In *Proceedings of The 9th Conference on Robot Learning*, volume 305, pp. 2714–2742. PMLR, 2025. URL https://proceedings.mlr.press/v305/saxena25a.html.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference (ESWC)*, pp. 593–607. Springer, 2018.

Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. MemoryVLA: Perceptual-Cognitive Memory in Vision-Language-Action Models for Robotic Manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 18567–18602, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/1fa3d6ccbcd15f7285fee666b2bd57be-Paper-Conference.pdf.

Yiheng Shu, Saisri Padmaja Jonnalagedda, Xiang Gao, Bernal Jiménez Gutiérrez, Weijian Qi, Kamalika Das, Huan Sun, and Yu Su. REMem: Reasoning with episodic memory in language agents. In *International Conference on Learning Representations*, volume 2026, pp. 55958–55987, 2026.

Sinan Tan, Mengmeng Ge, Di Guo, Huaping Liu, and Fuchun Sun. Knowledge-based embodied question answering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10): 11948–11960, 2023.

Qwen Team. Qwen2.5-VL Technical Report, 2025. URL https://arxiv.org/abs/2502.13923.

Ruiyang Wang, Hao-Lun Hsu, Swarajh Mehta, Jiwoo Kim, Zhihao Dou, and Miroslav Pajic. GAVEL: Graph world models for verified and efficient long-horizon LLM task planning. *arXiv preprint arXiv:2609.19315*, 2026.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations*, 2025.

Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10632–10643. IEEE, 2025a.

Yuncong Yang, Han Yang, Jiachen Zhou, Peihao Chen, Hongxin Zhang, Yilun Du, and Chuang Gan. 3D-Mem: 3D scene memory for embodied exploration and reasoning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17294–17303. IEEE, 2025b.

Shuo Yu, Yingbo Wang, Ruolin Li, Guchun Liu, Yanming Shen, Shaoxiong Ji, Bowen Li, Fengling Han, Xiuzhen Zhang, and Feng Xia. Graph2text or Graph2token: A perspective of large language models for graph learning. *ACM Transactions on Information Systems*, 44(3):1–49, 2026.

Shuang Zeng, Dekang Qi, Xinyuan Chang, Feng Xiong, Shichao Xie, Xiaolong Wu, Shiyi Liang, Mu Xu, and Xing Wei. JanusVLN: Decoupling Semantics and Spatiality with Dual Implicit Memory for Vision-Language Navigation. In *International Conference on Learning Representations*, volume 2026, pp. 33001–33026, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/3812c3c803e986eca64000fe92467991-Paper-Conference.pdf.

Jiahan Zhang, Muqing Jiang, Nanru Dai, Taiming Lu, Arda Uzunoglu, Shunchi Zhang, Yana Wei, Jiahao Wang, Vishal Patel, Paul Liang, Daniel Khashabi, Cheng Peng, Rama Chellappa, Tianmin Shu, Alan Yuille, Yilun Du, and Jieneng Chen. World-In-World: World Models in a Closed-Loop World. In *International Conference on Learning Representations*, volume 2026, pp. 55660–55699, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/5b4263be85820683d78675cc18d2efc7-Paper-Conference.pdf.

APPENDIX CONTENTS

Select a section title or page number to navigate to its supporting material.

Appendix	Section	Page
A	Frequently asked questions	16
B	Experiment map and evidence boundaries	17
C	Dataset construction and question schema	18
D	Architecture, representations, and runtime versions	22
E	Optimization and evaluation details	26
F	Hardware, software and compute	28
G	Human label audit	30
H	Statistical estimators and complete run results	34
I	Question-family results and sensitivity analyses	36
J	Graph optimization and correctness diagnostics	39
K	Validity probes, model panel, and evidence acquisition	41
L	Error anatomy: repairs, shared errors, and execution	43
M	Prompt catalog	48
N	Illustrative episodes and evaluated outcome examples	53
O	Development conditions and reproducible analysis	54
P	Extended related work	57
Q	Scope and transfer	60

A FREQUENTLY ASKED QUESTIONS

The title says knowledge-graph memory. What does that refer to, if the learned graph reader shows no gain? It refers to the record, not to the reader that failed. Both routes read the same scene graph; what differs is how it reaches the model. Writing it out as text is worth $+6.62$ points on MomaGraph-R1 and $+3.00$ on Gemma-3-4B. Encoding that same graph with a relational network and injecting the features gives no detectable gain on either backbone at either seed. We cannot say graph structure is what makes the text work. We can say the two readers behave differently, and finding that out meant building both.

Why might the learned graph reader not have helped? We have a hypothesis, not a demonstration. The node features do carry what the text route wins on: first and last sighting index, span, frequency. But one pathway averages across nodes before injecting anything, which blurs the per-node distinctions a question like "which did it see first" depends on. The other keeps nodes separate, yet enters after the final decoder layer behind a single scalar gate, where it can nudge an answer more easily than restructure a comparison. Both run together in every configuration we trained, so we cannot pin the null on pooling alone. Running them separately would settle it.

Could the graph arm simply be undertrained? We tested the most likely version of this. One graph-only run did fail to train, settling on almost the same answer letter every time. The gate schedule turned out to be the cause. We changed that one setting, and the fusion gate opened by more than an order of magnitude with the training diagnostic showing the pathway live (Figure 8). The recovered run repeats at a second seed and still does not beat its no-memory control. What this tests is recovery from that particular failure, not whether more or different training would help. The collapse was also specific: the combined text-and-graph arm converges under the short schedule on both backbones, as does Gemma's graph-only arm, though the combined arm's gate stays small and we did not train it under the longer schedule (Appendix J.1).

The headline is $+6.62$ points. Is that a large effect? Not as an average, and we do not present it as one. It is $+18.90$ points on the 291 route-history questions and -0.39 on the other 509, which average back to 6.63 over 800 against the 6.62 we report. Any explanation has to account for the null as well as the gain. That is why both groups appear throughout the paper rather than the aggregate alone.

The gain concentrates in three families. Are those families just exploitable? No: the gain sits where the priors fail. With no image at all, a model scores 6.5 points *below* chance on the route-history group and 23.2 points above chance on the set group (Figure 10). Put the other way, every set family can be answered above its own chance level from priors alone, and not one of them gains more than 2.7 points from memory. This is an association across families rather than a certificate for individual questions, and Order does carry exploitable options-only cues.

Why keep the two families your label audit flags? Because dropping them would hide the problem, and the result does not depend on them. Keeping them puts the scope limitation in front of the reader, and the reduced-set analysis then checks whether the direction survives without them. Removing Presence-change and Global-count takes out 222 of the 800 questions; on the remaining 578 the memory effect is $+5.80$ points on MomaGraph-R1 and $+3.46$ on Gemma-3-4B, both intervals excluding zero.

Why is the answer image blank? So that memory is the only thing that changes. If the model could see the room while answering, a gain might come from looking rather than from remembering, and a comparison between memory interfaces would tell us nothing. Every arm receives the same grey fill, so whatever separates them comes from the prompt and the graph.

Why is chance 33.0% and not 25%? Because the questions do not all have four options: 175 have two, 245 have three and 380 have four, which puts the question-weighted level at $\frac{1}{800}\sum_q 1/|\mathcal{C}_q| = 33.0\%$. It also varies by family. Order and Presence-change are binary at 50%, Counting is 28.6%. A single 25% baseline would flatter every result in the paper, and holding a binary family to 33% would do the opposite.

Why is the effect so much smaller on Gemma-3-4B? The direction and the structure replicate; the size does not, and we do not claim it does. Gemma gains $+3.00$ overall $[+0.69, +5.31]$, $+8.08$ on route-history families and $+0.10$ on set families, with graph memory null on both groups: the same dissociation at a smaller scale. With two backbones we cannot say whether capacity, pretraining or architecture accounts for the difference.

Why only two backbones and two seeds? Two seeds across two backbones is 16 supervised runs at roughly 12–29 hours each, plus the schedule diagnostics, sharing a node one GPU at a time. We spent that budget on matched contrasts rather than breadth: the same observations and the same step budget for every arm, with graph filtering and gate schedules disclosed. The cost is that our intervals cover test-episode sampling conditional on these checkpoints and say nothing about variation across new seeds. With two seeds they cannot.

Everything is simulated. Does this transfer to a real robot? We ran no physical robot and claim no transfer. The memory pipeline itself is not tied to the simulator: a prompted vision-language model reads RGB frames and writes scene-graph records, using no privileged state. The benchmark is tied to it, because the reference answers come from the simulator's per-frame visibility log, and reproducing those on real trajectories would need instance-level annotation. The ordering between interfaces could also change with perception quality, the scenes, or the answerer. Section 6 develops this.

Why not compare against existing embodied-QA methods? Because a fair comparison would mean matching observation budgets, training and answer-time access, and prior methods bundle memory together with their own perception, retrieval and exploration. Their published numbers cannot stand in as controls for an interface study. The generation protocol differs as well: procedurally sampled houses let us hold out splits that share no house with training and make route length a controlled variable at generation time rather than a property of whatever tours were recorded (Appendix Table 43). A system built to answer with the current view available has no memory interface to place in the A/B/C/D comparison at all. What we do report is nine off-the-shelf answerers on our questions with the cited evidence frames, each with its own image-free control, which shows how hard the questions are independently of our method (Table 2).

Is 800 questions enough? 800 is the held-out test split; the benchmark holds roughly 22,800 questions over 5,700 episodes, and the test episodes are house-disjoint from the rest. The main contrasts use a paired episode bootstrap that keeps each episode's four questions together, so the sample size is counted honestly. The reduced-set audit is reported separately and uses a question bootstrap (Table 11), a different sampling unit. The route-history group, where the effect lives, is 291 questions.

What would change your conclusion? A graph reader that keeps per-node temporal detail and enters early enough to use it — node-level attention without the pooled pathway, say, or injection at an intermediate decoder layer — beating the text route on route-history families at two seeds. Isolating the pooled pathway would test one explanation for why ours did not, though a different encoder or a different optimizer could change the ranking too.

B EXPERIMENT MAP AND EVIDENCE BOUNDARIES

This appendix organizes the available experiments by the information supplied to the model and the question each experiment addresses. It distinguishes current final-checkpoint results, diagnostic reruns, zero-shot evidence conditions, and superseded development runs. Their scores do not form a single ranking.

Table colors use the main-text scales throughout. Lavender numerical cells encode percentages from 0 to 100; sage and peach encode positive and negative effects with saturation at 30 percentage points. Correctness cells use sage for 1 and peach for 0. Headers use a uniform lavender tint. Numerical shading encodes magnitude; the reported intervals and protocol descriptions govern statistical interpretation and comparisons. All current supervised estimates use integer correctness exports from evaluations with blank answer images. Earlier development runs differ in split size, visual inputs, or

Table 3: Experimental families. Every current test condition designates 200 episodes and 800 questions unless its coverage is explicitly qualified.

Family	Experimental question	Evidence and boundary
Supervised A/B/C/D	Which memory interface helps a trained answerer?	Same blank answer image; matched seed counts; C/D graph filtering differs.
Long gate warmup	Can graph-only optimization recover?	Retrained MomaGraph C, seed 0; longer scaffold schedule.
Untuned memory control	Does supplying text help without adaptation?	Same labeled questions; launch provenance is less complete.
Family/group analysis	Which questions account for the gain?	Paired correctness records; exploratory grouping and sensitivity checks.
Correctness flips	Which items change across arms?	Correct/incorrect outcomes, not answer-letter identity or explanations.
Blind/options probes	How much is answerable without images?	Qwen3-VL-4B and wider probe panel; family rates are reported aggregates.
Evidence ladder	Does access to evidence differ from acquisition?	Matched Qwen3 backbone; oracle images, replay, and closed-loop exploration.
Zero-shot model panel	How do contemporary answerers perform?	Different evidence adapters and parse rates; no supervised-arm comparison.
Development runs	How did the protocol evolve?	Historical diagnostics only; obsolete settings are not additional replications.

optimization settings and are excluded from those estimates. Appendix O specifies this separation and the steps needed to reproduce the prediction analyses.

C Dataset construction and question schema

C.1 Family grouping

The eight question families. The descriptive *route-history* group concerns sightings and room visits; the *set* group concerns object presence, location, and counts. Table 1 gives definitions, abbreviations, and test counts.

C.2 Intended memory demand

Intended memory demand and observed validity. Evidence references identify observations used when constructing a question. The `memory_required` flag checks whether the latest referenced frame occurs before termination and has a room label different from the final room. This geometric heuristic and the family labels describe intended evidence use; image-free controls assess answerability from priors. We therefore retain the full test population and report image-free controls. For exploratory analysis, we group Order, Presence-change, and Rooms-visited as *route-history families* (291 questions), with the other five families forming a 509-question comparison group. The grouping collects questions about sightings and visited rooms; it does not identify a common temporal reasoning operation. Counting distinct visited rooms, for example, need not use visit order.

C.3 Evaluation-population details

Evaluation population. The reported training and validation splits contain 5,000 and 500 episodes. The test set contains 200 episodes with four questions each, balanced across the four route strata, at 200 questions per stratum. Cache logs record 236,899 training, 21,482 validation, and 8,135 test frames. Here, long-horizon refers to retaining observations across a trajectory, in-

cluding multi-room routes. Whether memory benefit grows with route length is a separate empirical question. The three splits are disjoint at the level of houses: no house contributing episodes to the test split contributes episodes to training or validation. We verify this by object identity rather than by the `house_idx` field, whose values are assigned per split and therefore collide across splits without denoting a shared house; pooling the object instances of each house, colliding indices overlap no more than unrelated house pairs do (Appendix H). The test set has 175 two-option, 245 three-option, and 380 four-option questions, giving a question-weighted chance level

$$a_{\text{chance}} = \frac{1}{800} \sum_{q} \frac{1}{|\mathcal{C}_q|} \approx 0.3302. \tag{6}$$

We report this as 33.0%. A constant answer letter is a separate baseline, determined by the label distribution. Family counts and generation details appear in Appendix C.

C.4 Validation details

Validation checks. Validation runs in two stages with different purposes. Stage A applies generation-time filters: a target-mention filter removes questions that name the navigation target, a symbolic checker re-derives each answer from the per-frame visible-object log, and contradictions are dropped rather than repaired. Stage B applies post-hoc probes to the *released* split and is deliberately not a filter. A blind probe supplies the question and options with no images; an options-only probe supplies the options with no question text. Both run per family, and their role is to quantify how much of each family is answerable without the evidence it nominally requires, rather than to remove those questions. We therefore retain the probe-sensitive questions in the test population and report the probes alongside the results. Four of the eight families are probe-sensitive on this audit, covering 397 of the 800 test questions.

C.5 Question-writing details

Question writing. Questions are written by symbolic templates over the episode record rather than by a language model: each template reads the per-frame visible-object log and emits both the question and a reference answer with its justification, so every item in the released set is traceable to a rule rather than to a generation. Up to six frames are sampled across the trajectory as the evidence strip, and `frame_refs` are remapped from strip-local to episode-local indices. Each question carries two to four options with the option order shuffled, so the correct letter is not positionally informative. Eight families cover Room Identification, Existence, Counting, Global-count, Persistence, Order, Presence-change, and Rooms-visited; Table 1 defines each and gives the abbreviation used in later tables and figures. Some family names have specific operational meanings. Order compares first sightings of object types; Persistence counts the room labels in which a type was observed; the reference Presence-change rule checks earlier visibility followed by absence at the final sampled frame of the strip, which is narrower than a general notion of disappearance. The target navigation object and room are withheld from the question-generation prompt.

C.6 Episode-generation details

Episode generation. We use ProcTHOR-10k houses (Deitke et al., 2022) in AI2-THOR (Kolve et al., 2017), under the AI2-THOR 5.0.0 controller. The agent acts in a discrete space of six actions: `MoveAhead`, `RotateLeft`, `RotateRight`, `LookUp`, `LookDown` and `Done`. Each episode is written as one `ep_*.json` record together with the 256×256 RGB frames it references. The generator separates simulator traversal from question construction , so no question can influence the route that produced it. Route strata S1–S4 have zero, one, two, or at least three door crossings, respectively, spanning same-room observations to multi-room histories. A door crossing is one hop in the breadth-first door graph of the house, and we use *door crossing* and *hop* interchangeably for that quantity. Episode records include actions, visible objects, room labels, and evidence-frame references , together with navigation outcomes for the traversal itself: whether the agent halted within a fixed stopping distance of its target with that target verified present by instance segmentation, and the walked path length against the geodesic distance available at spawn. These describe the route that produced an episode rather than the task studied here. We release them because they document how each trajectory was obtained, but no result in this paper is computed from them, every number we report scores question answering after the traversal has ended.

C.7 CONSTRUCTION OVERVIEW

Figure 3 summarises construction in four stages, described in turn below.

The generator runs simulator traversal and question construction as separate stages. The traversal stage samples a target and spawn state, plans through a room graph, executes actions, and writes observations and actions. The question stage samples a chronological strip and creates multiple-choice questions. This allows question revisions without repeating traversal.

C.8 ROUTE STRATA AND EPISODE RECORDS

The reference configuration defines S1 through S4 by zero, one, two, or at least three door crossings. This is a route-length design variable, not a statement that every test item needs the full route. The recorded episode schema includes the house index, room sequence, frame paths, action sequence, target object instance, termination reason, and navigation measures. Per-frame metadata includes visible objects, room identity, and the action executed. Question records include text, options, reference letter, category, family, reasoning type, evidence references, and the memory-required flag. The cache totals of 236,899/21,482/8,135 frames for train/validation/test describe available observations, not answer-prompt tokens or the number of views individually recoverable from a truncated memory. These totals describe observation coverage; measuring evidence-to-answer lag additionally requires item-level evidence indices.

The reference generator supports logging rejected attempts with failure reasons and distinguishing discrete trajectories from routes using a teleport fallback. The evidence-ladder exports contain 50 test episodes in each stratum (Table 27). We use these exported strata for route analyses.

C.9 QUESTIONS, SYMBOLIC METADATA, AND LABEL LIMITS

Table 4: Current test family counts. The grouping is used only for exploratory analysis.

Family	Questions	Analysis group
Order	73	Route-history
Presence-change	102	Route-history
Rooms-visited	116	Route-history
Counting	65	Other
Existence	113	Other
Global-count	120	Other
Persistence	131	Other
Room Identification	80	Other
Total	800	200 episodes

The available question-generation implementations combine model-generated candidates with fallback templates and symbolic checks. The prompt withholds the designated target object and target room, requests references to supporting observations, and discourages explicit frame numbers in question text. Schema validation checks that questions have two to four distinct option strings, a valid answer label, and evidence references. Cross-frame reasoning types require multiple references. The code also flags whether question text names the navigation target or its room.

These checks validate record structure and selected target-disclosure channels. Visual entailment and object identity are separate annotation questions. The reported accuracies score the stored reference labels; the analyzed exports contain question identifiers, families, and correctness. Appendix N distinguishes these evaluated outcomes from full illustrative questions drawn from a separate training collection. The reference parser can also replace absent or invalid evidence references with a default first-frame reference before schema validation. A populated reference field can therefore reflect a parser default; content verification requires checking the cited observations.

Presence-change is an endpoint criterion. The patched reference generator labels a previously observed object type as disappeared when its last sampled sighting precedes the final frame. This tests terminal absence, not whether any intervening disappearance occurred. The wording *visible*

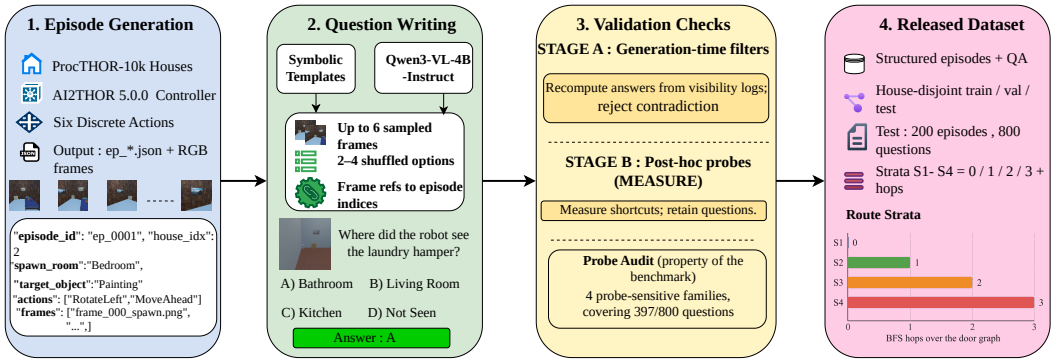
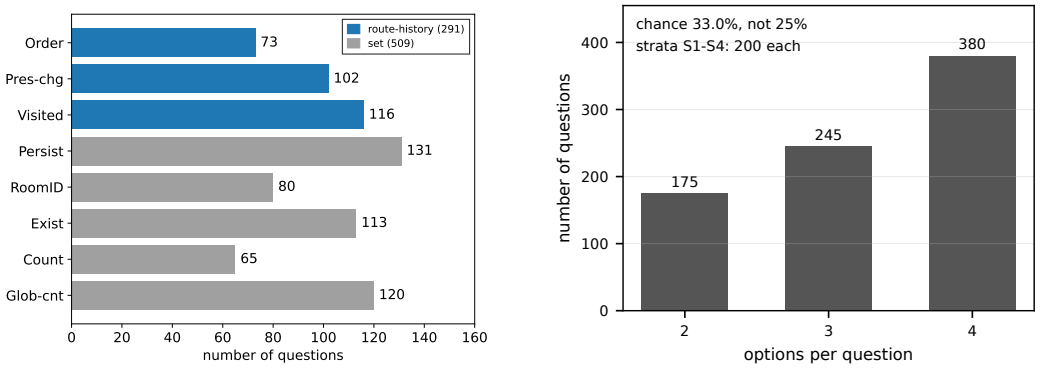


Figure 3: **Dataset construction.** Episode generation in ProcTHOR/AI2-THOR is followed by question writing (symbolic templates over the episode record), a two-stage validation pass (generation-time filters, then post-hoc blind and options-only probes that *measure* rather than remove leakage), and the released split. Route strata S1–S4 denote zero, one, two, or three-or-more door crossings; they are a route-length variable, not a validated difficulty ordering (Appendix C). Four of eight question families are probe-sensitive, covering 397 of 800 test questions (Appendix K).



(a) Questions per family.

(b) Option counts per question.

Figure 4: **Benchmark composition.** 291 route-history against 509 set questions (a); the mix of two-, three- and four-option questions puts question-weighted chance at 33.0%, not the 25% a uniform four-option assumption would give (b).

at some point but then gone from view later can admit the broader interpretation, including disappearance followed by reappearance. The reported family label is retained when scoring the existing predictions, but an episode-level audit is needed to determine whether this ambiguity affects current test items. We do not infer label error for a particular question from the template alone. Moreover, leaving the camera's view does not imply a physical change in the environment.

C.10 Question families and memory-required metadata

The reference `memory_required` predicate is true when the latest referenced frame precedes the final frame and the room at that evidence frame differs from the final room. It is a geometric/evidence heuristic. It can be false for a question that needs earlier observations within one room, and true for a question predictable from nonvisual priors. We do not use it as a gold label for cognitive memory demand or claim that memory gains concentrate on this flag. Figure 3 in Section 3 gives the end-to-end construction pipeline, including the post-hoc leakage probes discussed in Appendix K.

Table 5: Reference-generation targets and production-specific distinctions. Room labels and physical room instances represent different counting units.

Family	Reference target and qualification
Room Identification	The room label associated with sightings of a queried object type.
Existence	Whether an object type was observed in a named room.
Counting	Distinct simulator object IDs of a type observed in a named room; the memory's type counts are an approximation.
Persistence	Number of distinct room labels in which an object type was observed, rather than uninterrupted visibility.
Order	Relative first-sighting indices of two object types.
Presence-change	Earlier observation with absence at the final sampled frame; not every disappearance/reappearance event.
Global-count	Counts over the sampled evidence strip; wording can imply whole-house coverage (Appendix G).
Rooms-visited	A current exported family; whether each question counts room instances, labels, or visits requires its production record. No order requirement is assumed.

D Architecture, representations, and runtime versions

D.1 Representation and reader definitions

Architecture details. **From observations to memory-conditioned answers.** The combined configuration D exposes the episodic record as text and as learned graph features. The graph encoder supplies two residuals: pooled input injection (gate $\tanh(\alpha)$, pathway A) and node-level output attention (gate $\tanh(\beta)$, pathway B, after the decoder). Navigation uses real images at each step; answering uses a blank image at Done. The same decoder supports both heads. A/B/C disable the corresponding memory interfaces in Appendix Table 6. MomaGraph-R1's decoder has 28 layers and 88.7M trainable parameters; Gemma-3-4B has 34 layers, $d = 2560$ and 52.5M trainable parameters, and is otherwise identical.

Text readout. For room label r and object type c, let $n_t(r, c)$ count retained occurrences of type c in room r at frame t. The sighting set and summary are

$$J_{r,c} = \{t : n_t(r, c) > 0\}, \qquad \mu_{r,c} = \big(\min J_{r,c}, \max J_{r,c}, |J_{r,c}|, \max_t n_t(r, c)\big). \tag{7}$$

For each observed type, text reports first and last sightings and maximum simultaneous count. Frequency prioritizes entries under the budget of 12 types per room; at most 24 relations are retained. Consecutive equal room labels are compressed into visits with frame spans. Thus, a return visit remains explicit even when objects are grouped under the same room label. A record such as `Bed[0-1]` reports endpoints, not continuous visibility.

Graph features. The schema has 104 object types, four room types, and 12 relation types, giving a node vocabulary of 108 entries. Each node starts with $x_v \in \mathbb{R}^{85}$: a learned 64-dimensional name embedding, one room/object indicator, six state flags, eight function flags, and six temporal features. The reference converter initializes state and function fields as empty. For a node's sighting indices J_v, the temporal block records normalized first and last indices, their span, sighting frequency, initial presence, and final-frame presence. These features summarize history at a node; relations encode the accumulated spatial and room structure.

Relational graph reader. The text path inserts $\mathcal{R}(M)$ into the answer prompt alongside the question and full option strings. The graph path learns a reader for G. A relational graph convolution (Schlichtkrull et al., 2018) updates node states as

$$h_v^{(\ell+1)} = \mathrm{ReLU}\left(\mathrm{LN}\left[h_v^{(\ell)} W_0^{(\ell)} + \sum_{k\in\mathcal{K}} \frac{\sum_{u\in\mathcal{N}_k(v)} h_u^{(\ell)} W_k^{(\ell)}}{|\mathcal{N}_k(v)|} + b^{(\ell)}\right]\right), \tag{8}$$

Here $h_v^{(0)} = x_v$, $\mathcal{K}$ indexes relation types, and $\mathcal{N}_k(v)$ contains neighbors of v under relation k; empty neighborhoods contribute zero. Two layers map $85 \to 128 \to 256$ dimensions; projection gives $H_G = \mathrm{LN}(H^{(2)} W_p + b_p) \in \mathbb{R}^{|V|\times d}$. The shared vision-language model receives graph features before and after its decoder:

Decoder states.

$$H = \mathrm{Decoder}_{\theta_0+\Delta\theta}\Big(\mathrm{pack}(\mathrm{Vis}(I), \tilde{E})\Big). \tag{9}$$

Fusion and backbone details. Here E contains token embeddings, which are combined with visual embeddings for the decoder; $H \in \mathbb{R}^{L\times d}$ contains the resulting sequence states. $\mathrm{Vis}(I)$ supplies image tokens for the real or blank image. The pooled residual broadcasts over token embeddings; output attention accesses individual nodes. LN is layer normalization; MHA is eight-head cross-attention. MomaGraph-R1 (Ju et al., 2026) uses $d = 3584$ and 28 decoder layers; Gemma (Gemma Team, 2025) uses $d = 2560$ and 34 layers. Base weights θ_0 are frozen; $\Delta\theta$ denotes LoRA updates on W_q, W_k, W_v, W_o, giving 88.7M trainable parameters of 8.38B total for MomaGraph-R1 and 52.5M for Gemma. The pooled residual of Eq. (2) is gated per channel and the output residual of Eq. (3) by a scalar. The learned input gate initializes to 0.02 and the output gate to zero. During training, a temporary gate offset $b(s)$ rises linearly from 0 to 0.2 over w optimizer steps and falls to 0 over the next $2w$ steps. Inference sets $b(s) = 0$. The offset applies to the output gate of Eq. (3) only; the input gate of Eq. (2) is never biased. This graph schedule is separate from learning-rate warmup, which runs concurrently over the first 300 steps (Section 4.3).

D.2 Memory configurations

The architecture is shown in Figure 1. Real and blank image icons there distinguish navigation and answer inputs. Displayed backbone dimensions correspond to MomaGraph; Gemma dimensions and exact parameter counts are given in this appendix.

D.3 Observation sources and node identity

The scene-graph prompt asks for visible entities, relations, room type, and task context. The current-room field `scene_type` and task-implied `target_scene` are separate. The visible parser uses the former for the observed room. This separation avoids directly substituting a desired room for an observed room, but it does not eliminate all task-context bias from perception. The reference path conditions perception on the first episode question. When the inferred current and target rooms differ, its prompt requests only navigation structures. This task-conditioned filtering can suppress otherwise visible objects and can favor the first question. The checkpoint, decoding settings, and cache identity are therefore part of the configuration needed to reproduce memory construction.

The reference converter retains spatial relationships and initializes object state and function fields as empty. It uses synthetic per-frame object identifiers and room-type labels rather than geometric instance tracking. Runtime arguments select a canonical node-key mode. These identity and feature conventions determine which distinctions the accumulated record can represent.

Table 6: **Memory configurations.** All receive a blank answer image, question, and choices. Architectural nodes include walls and doors; text filters these entities by default. Graph access enables both residual pathways.

Config.	Text	Graph	Architectural graph nodes
A	No	No	–
B	Yes	No	–
C	No	Yes	Dropped
D	Yes	Yes	Retained

D.4 TEXT REPRESENTATION

The text renderer summarizes the ordered predicted frame records independently of the encoded graph tensor. Its reference output uses the following template:

```
ROOMS: {observed room labels}
VISITS (frames f0-fT): {room}@{first-last} -> ...
{room}: {object type}[{first-last}], {object type}
↪  x{count}[{first-last}], ...
RELATIONS: {type}-{relation}-{type}; ...
```

The reference renderer limits objects per room to 12 and relations to 24. These are object and relation budgets, not a guaranteed fixed token count. Objects are prioritized using sighting frequency and first occurrence after per-frame truncation. Architectural objects are dropped from the answer-memory text by default. The multiplicity attached to an object type is the largest simultaneously visible count, so it can underestimate distinct instances encountered at different times.

An interval such as `[f0-f3]` means the first and last recorded sightings occurred at those indices. It does not assert continuous visibility through the interval. A return to a room can also involve a new physical instance of the same object type. These representational limitations matter for disappearance and global-count questions.

D.5 GRAPH SCHEMA AND TEMPORAL FEATURES

The current vocabulary contains 104 object types and four room types. Runtime headers report 12 graph relation types. The 108 name categories are represented within an 85-dimensional input feature vector. This vector contains a learned 64-dimensional name embedding, one room/object indicator, six state flags, eight function flags, and six temporal features. State and function slots use the converter defaults described above.

Let J_v denote the frame indices at which node v is observed among T retained frames, with $d = \max(1, T-1)$. The temporal block is

$$\left[\frac{\min J_v}{d}, \frac{\max J_v}{d}, \frac{\max J_v - \min J_v}{d}, \frac{|J_v|}{T}, \mathbf{1}\{\min J_v < \max(1, n_{\text{spawn}})\}, \mathbf{1}\{\max J_v \geq T-1\}\right]. \tag{10}$$

It represents first sighting, last sighting, observed span, frequency, initial presence, and latest-frame presence. It does not preserve every intermediate appearance transition. Consecutive predicted room labels induce adjacency edges, which should not be interpreted as a verified architectural floor plan.

D.6 TWO GRAPH PATHWAYS

The reference encoder applies two relational graph convolution layers (Schlichtkrull et al., 2018), $85 \to 128 \to 256$, with normalization and ReLU. A linear projection and LayerNorm map node features to backbone width. The runtime widths are 3,584 for MomaGraph-R1 and 2,560 for Gemma.

The input pathway averages non-padding projected node embeddings, transforms the mean through an MLP and normalization, gates it elementwise, and adds the same residual at token embedding outputs. The gate is a vector initialized to 0.02. No additional graph token positions are required by

this route. If Z denotes the projected nodes,

$$E' = E + \tanh(\boldsymbol{\alpha}) \odot \mathrm{LN}(\mathrm{MLP}(\mathrm{mean}_v Z_v)) \,. \tag{11}$$

The output pathway uses eight-head cross-attention to the unpooled nodes after the decoder:

$$H' = H + \big(\tanh(\beta) + b(s)\big)\mathrm{MHA}(\mathrm{LN}_q(H), \mathrm{LN}_{kv}(Z), \mathrm{LN}_{kv}(Z)) \,. \tag{12}$$

The scalar β initializes to zero. The training-only scaffold $b(s)$ is zero at inference. A zero gate therefore preserves the residual input in the absence of the scaffold.

The encoder transforms temporal features before pooling, and output cross-attention retains access to individual nodes. Attributing the graph result to pooling would require separate pathway and pooling ablations.

D.7 Formal memory summaries and readout

For room label r and object type c, let $n_t(r, c)$ count their occurrences in the retained records of predicted frame S_t, after per-frame filtering and truncation. Define the retained sighting set and text summary by

$$J_{r,c} = \{t \in \{0, \ldots, T-1\} : n_t(r, c) > 0\}, \tag{13}$$

$$\mu_{r,c} = \left(\min J_{r,c}, \max J_{r,c}, |J_{r,c}|, \max_{0 \le t < T} n_t(r, c)\right) . \tag{14}$$

The text displays each type's first and last sighting and its maximum simultaneous count. When entries exceed the text budget, the renderer prioritizes frequently observed types. Types with no retained sightings are omitted. Let ρ_t be the predicted room label. Compressing consecutive equal values in $(\rho_0, \ldots, \rho_{T-1})$ produces the visit sequence, with the corresponding frame spans. Consequently, returning to the same room type can produce another visit in the text even though its objects are grouped under the same room label.

These summaries encode specific temporal statistics. For example, sighting sets $\{0, 1, 3, 4\}$ and $\{0, 2, 3, 4\}$ have equal first index, last index, frequency, span, and endpoint-presence flags. Their per-node temporal blocks are identical although the intervening visibility sequence differs. This is a representational example, not a measured error. Similarly, with ideal instance-identified observations $O_t(r, c)$,

$$\max_t |O_t(r, c)| \;\le\; \left|\bigcup_t O_t(r, c)\right| . \tag{15}$$

The left side summarizes simultaneous visibility, while the right side counts distinct instances across time. Predicted type counts approximate the former; Equation 15 is an inequality for ideal observations, not a calibrated bound on a perception model's predictions. This distinction motivates evaluating instance-aware memory alongside the type-level interface.

Relational encoding. Writing node states as row vectors, a reference relational layer has the form

$$h_v^{(\ell+1)} = \mathrm{ReLU}\left(\mathrm{LN}\left(h_v^{(\ell)} W_0^{(\ell)} + \sum_{r \in \mathcal{R}} \frac{1}{|\mathcal{N}_r(v)|} \sum_{u \in \mathcal{N}_r(v)} h_u^{(\ell)} W_r^{(\ell)} + b^{(\ell)}\right)\right) , \tag{16}$$

where empty neighborhoods contribute zero. The two layers use widths $85 \to 128 \to 256$; W_0 is the root transform and the relation-specific W_r mix typed neighboring nodes. The projected states $H_G = \mathrm{LN}(H^{(2)} W_p + b_p)$ are shared by pooled input injection and node-level output attention. Downstream losses train the encoder; a separate prompted model extracts scene graphs.

Conditioning the four configurations. Let b_{text} and b_{graph} be binary switches. Their values for A, B, C, and D are $(0, 0)$, $(1, 0)$, $(0, 1)$, and $(1, 1)$. The answer prompt contains

$$x_j = \big(I_{\mathrm{blank}}, q_j, \mathcal{C}_{q_j}, \mathrm{slot}(b_{\mathrm{text}}, \mathcal{R}(M))\big), \tag{17}$$

Here $\mathcal{C}_{q_j}$ contains the full option strings (letters and answer texts), while $\mathcal{O}_j$ contains their answer-letter token IDs. The slot contains memory text only when enabled. Both graph residuals are bypassed when $b_{\mathrm{graph}} = 0$. The switches specify the interfaces; the configuration-specific filtering in Table 6 still applies, so they do not imply identical retained text and graph content.

If $y_{j,1:K_j}$ is the tokenized target answer and $\mathcal{V}$ the language vocabulary, the reference answer loss is

$$\mathcal{L}_{\text{ans},j} = -\frac{1}{K_j}\sum_{k=1}^{K_j}\log\frac{\exp \ell_{j,k}(y_{j,k})}{\sum_{u\in\mathcal{V}}\exp \ell_{j,k}(u)}. \tag{18}$$

The prompt tokens are masked from this loss; target positions use autoregressive context. At evaluation the candidate distribution instead normalizes next-token logits over the provided option letters,

$$\pi_j(u) = \frac{\exp \ell_j(u)}{\sum_{v\in\mathcal{O}_j}\exp \ell_j(v)}, \quad u \in \mathcal{O}_j, \qquad \hat{y}_j = \arg\max_{u\in\mathcal{O}_j}\pi_j(u). \tag{19}$$

Restricted evaluation scoring preserves the candidate ranking but is distinct from the full-vocabulary training loss. The action loss is $-\log p_\theta(a_t \mid I_t, S_t, \text{question context, enabled graph context})$, with six candidate actions. Equation 4 averages answer losses only on Done samples.

D.8 Navigation and answer supervision

The navigation head reads a designated classification token and predicts one of six actions: MoveAhead, RotateLeft, RotateRight, LookUp, LookDown, or Done. Its prompt contains the real current image, local scene-graph text, candidate actions, and question context. The reference answer head is the pretrained language-model output head. Training applies cross-entropy to the target answer letter over the full vocabulary; evaluation constrains candidate scoring to the letters provided in the question. It is not a separately trained four-way answer classifier.

Navigation supervision applies at valid frames. Answer supervision applies on Done samples, averages over the episode's questions, and uses weight $\lambda = 1$ in the recorded runs. Done samples are oversampled eightfold. Base model weights are frozen while LoRA (Hu et al., 2022) and additional instantiated modules are trainable. The reported parameter totals include graph-side modules even in configurations that bypass them, so these are not per-arm counts of parameters receiving gradients.

D.9 Reference implementation versus deployed protocol

Archived runtime arguments and the evaluator patch specify the reported blank-answer-image protocol. The reference trainer represents an earlier version with real answer images; the deployed runs also use text-only flags and Gemma loading behavior beyond that version. The equations describe the reference architecture; experimental settings follow runtime logs.

The prediction exports support recomputation of accuracy and paired contrasts. Reproducing upstream construction also requires matching the production code, perception cache, episode records, and split manifest to those exports. The archived reference material provides only a partial mapping across these artifacts, so production-specific question semantics and runtime details require the corresponding versioned sources. Figure 1 specifies the observation and answer-time inputs.

E Optimization and evaluation details

E.1 Supervision and scoring details

Graph conditioning is bypassed in A/B. The pretrained language-model head maps $\tilde{H}$ to answer logits. For target tokens $y_{j,1:K_j}$ and language vocabulary $\mathcal{V}$, prompt positions are masked and

$$\mathcal{L}_{\text{ans},j} = -\frac{1}{K_j}\sum_{k=1}^{K_j}\log\frac{\exp \ell_{j,k}(y_{j,k})}{\sum_{u\in\mathcal{V}}\exp \ell_{j,k}(u)}, \qquad \hat{y}_j = \arg\max_{u\in\mathcal{O}_j}\ell_j(u). \tag{20}$$

Training uses full-vocabulary cross-entropy; evaluation compares next-token logits only for the supplied answer-letter IDs $\mathcal{O}_j$. The full option texts remain in the prompt. The graph encoder and fusion modules learn from the joint loss when enabled; scene-graph extraction remains a separate prompted step. Runs use $\lambda = 1$, eightfold Done oversampling, and LoRA (Hu et al., 2022) with rank 16, scale 32, and dropout 0.05, applied to the attention projections only and leaving bias terms untrained. Learning rate is 10^{-4}, effective batch size 16 (batch size 1 with 16 gradient accumulation steps),

weight decay 0.01, with cosine decay, 300 optimizer warmup steps (5% of the step budget), and 6,000 steps per configuration. Early stopping is disabled, so 6,000 steps is a fixed budget rather than a validation-selected stopping point, and main results evaluate the final checkpoint; no result in this paper uses a validation-selected checkpoint. Each configuration trains on a single NVIDIA H200 (140 GiB) in bfloat16, with one card per run and no model, pipeline or data parallelism. Additional feature definitions and runtime settings appear in Appendices D–F, the last of which reports per-arm throughput, memory footprint, and total compute including the cost of generating the dataset.

Table 7: Reported training settings, taken from the runtime logs of the released checkpoints.

Setting	Reported value
Primary backbone	MomaGraph-R1 (Ju et al., 2026), Qwen2.5-VL-based (Team, 2025)
Secondary backbone	Gemma-3-4B-IT (Gemma Team, 2025)
Numerical precision	bfloat16
LoRA targets	W_q, W_k, W_v, W_o
LoRA rank / alpha / dropout (reference)	16 / 32 / 0.05
Batch size / accumulation	1 / 16
Maximum optimizer steps	6,000
Learning rate / decay	10^{-4} / cosine
Optimizer warmup	5% of budget (300 steps)
Minimum learning-rate ratio	0.05
Weight decay / Adam epsilon	0.01 / 10^{-8}
Adam betas	0.9, 0.999
Gradient clipping	1.0
Answer loss weight / Done oversampling	1.0 / 8
Validation interval	500 optimizer steps
Online answer probe	100 episodes, approximately 400 questions
Full validation evaluation	500 episodes, 2,000 questions
Final test evaluation	200 episodes, 800 questions
Checkpoint used for main results	Final fixed-budget checkpoint
Original graph-gate warmup	400 steps
Recovered MomaGraph C warmup	2,000 steps
Reported trainable count (MomaGraph)	88,756,235 instantiated parameters
Reported trainable count (Gemma)	52,523,019 instantiated parameters

The answer-image correction is part of the evaluation contract. Earlier evaluation logs exposed real answer images despite training with blanks and are superseded for the supervised comparison. The same checkpoint produces different accuracy under the two input conditions; this is an input-distribution mismatch, not evidence of a new model improvement. We keep the corrected evaluations separate from these historical records.

The graph-gate scaffold increases from zero to 0.2 over w optimizer steps and then decreases to zero over the next $2w$ steps. With $w = 400$, the scaffold disappears at step 1,200; with $w = 2000$, it disappears at the final step, 6,000. It is not the optimizer's learning-rate warmup. Validation and final test evaluation use the learned gate without the scaffold. The recovered final fusion parameter is negative (approximately -0.0560); the diagnostic plot displays its magnitude, so the signed parameter is needed to interpret the residual direction. A longer training period after scaffold removal would provide a cleaner test of sustained optimization stability than the present run.

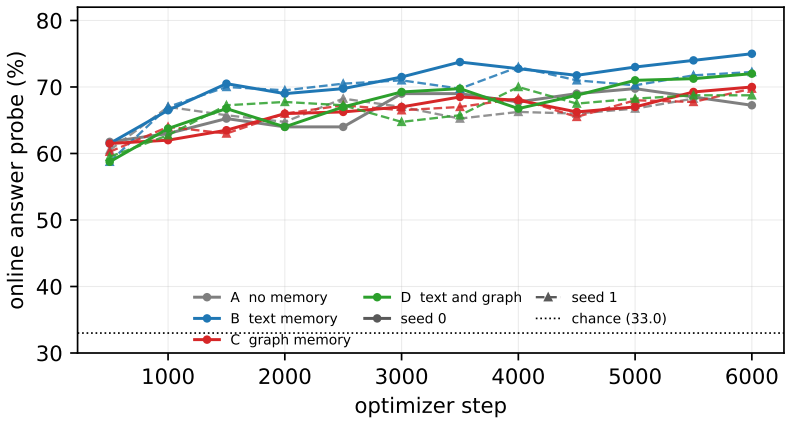


Figure 5: **The reported ordering is not an artifact of where training stopped.** Online answer probe during training, four MomaGraph-R1 configurations at two seeds each; solid lines are seed 0 and dashed lines seed 1. The graph-memory arm is the corrected-schedule run. Early stopping is disabled and every run trains the full 6,000-step budget, so this is a monitoring signal and not a selection criterion; the text configuration separates by roughly step 1,500 and stays separated at both seeds. **These are not the reported accuracies**: the probe covers about 400 questions over 100 episodes and reads several points higher than the held-out test split.

F Hardware, Software and Compute

F.1 Hardware

All training and evaluation ran on a single shared node. Each configuration was trained on **one NVIDIA H200 GPU with 140 GiB of device memory**, with no model, pipeline or data parallelism: one run occupies one card for its entire duration, and concurrent runs were placed on separate cards. The node carries eight H200 cards, two AMD EPYC 9575F processors (64 cores each, 128 physical cores in total) and 2.2 TiB of system RAM, using local NVMe-backed XFS for data and checkpoints.

Placing one run per card rather than sharding a larger model is what makes the per-arm wall-clock figures in Table 8 comparable: no run competes with another for memory bandwidth on the same device. No configuration approaches the card's capacity (the largest reserved footprint we observed was roughly 70% of one H200), so the 140 GiB card is not the binding constraint on batch size; the effective batch of 16 comes from gradient accumulation at batch size 1 to keep prompt-length variation from changing the optimization, not from a memory limit.

F.2 Software

Component	Version
Python	3.12.13
PyTorch	2.11.0+cu130
Transformers	5.7.0
PEFT	0.19.1
NVIDIA driver / CUDA runtime	590.48.01 / 13.0
Simulator	AI2-THOR 5.0.0 (Kolve et al., 2017)
Numerical precision	bfloat16 (autocast and model weights)

One environment serves every supervised run and every zero-shot panel evaluation. This is deliberate: a second, differently pinned environment would make the zero-shot panel a comparison across software stacks as well as across models. One model in that panel, InternVL2.5-8B, does not load under this Transformers version because its vendored modelling code targets an older API. We report it as a version incompatibility rather than creating a second environment for it (Appendix K).

F.3 TRAINING COST

Every configuration trains for a fixed budget of 6,000 optimizer steps at an effective batch size of 16 (batch size 1 with 16 gradient accumulation steps). Throughput differs across arms for two reasons: the configurations carrying the rendered record put roughly 1.2k additional characters in each prompt, and the graph configurations run the relational encoder and the fusion modules in addition to the backbone.

Table 8: Observed throughput and wall-clock for 6,000 steps on one H200. **These are wall-clock times on a shared node and include contention**: the Gemma runs overlapped one another and other users' jobs, and the same configuration on an otherwise idle card sustains roughly 900 steps/hour. The figures describe these contended runs, not peak hardware throughput.

Configuration	Steps/hour	Hours for 6,000 steps
MomaGraph-R1 (8.4B total, 88.7M trainable)		
A, no memory	490	12
B, text memory	365	17
C, graph memory	550	11
D, text and graph	320	19
Gemma-3-4B-IT (52.5M trainable)		
A, no memory	218	27.6
B, text memory	205	29.3
C, graph memory	238	25.2
D, text and graph	235	25.5

We did not instrument memory per configuration. Sampling the two runs that were still training when we checked, the caching allocator had reserved 97 and 84 GiB of the card's 140 GiB, so no configuration came close to filling it. Batch size is therefore not memory-constrained: the effective batch of 16 comes from gradient accumulation at batch size 1, chosen to keep prompt-length variation from changing the optimization.

F.4 DATASET GENERATION

Generating the benchmark cost substantially more compute than training on it, and we report it because a reader reproducing this work from scratch pays that cost first. Generation has three stages with very different profiles.

Episode collection. 5,700 episodes are executed in AI2-THOR, each navigating a ProcTHOR (Deitke et al., 2022) house under the discrete action set and writing its 256×256 frames to disk. This stage is simulator-bound rather than GPU-bound: the cost is scene loading, physics stepping and rendering, and it parallelises across processes but not across GPUs. It produced 266,516 retained frames across 5,700 episodes; wall-clock for this stage was not instrumented.

Scene-graph extraction. A prompted vision-language model runs over *every retained frame* of every episode to produce the per-frame records from which both memory routes are built. This is the dominant cost of the whole project. The cached splits contain 236,899 training, 21,482 validation and 8,135 test frames (**266,516 forward passes of a vision-language model**), and the cache is built once and reused by every subsequent training run, which is why the caches are a released artifact rather than something each run recomputes. Wall-clock for this stage was not instrumented, but its scale is the 266,516 forward passes above, which is roughly two orders of magnitude more model invocations than a single training run performs.

Question generation and validation. Symbolic templates and a prompted VLM write the questions, after which the Stage A filters re-derive each answer from the per-frame visibility log and drop contradictions. Cost here is borne by the symbolic checks rather than by any model, since the released questions are template-generated; this cheaper stage was not timed separately.

Why this matters for the reported totals. The training figures in Table 8 assume the scene-graph cache already exists. Training all seventeen reported runs from a cold start would additionally

require the extraction stage above, once. We state the two separately rather than combining them, because they are incurred once and per-run respectively.

Total compute. The reported study comprises 17 supervised training runs: four configurations $\times$ two seeds on each of two backbones, plus one additional graph-only run at a shorter gate schedule that failed to train and is analysed in Appendix E. At the observed wall-clock above this is approximately 344 GPU-hours, of which roughly 215 are the eight Gemma-3-4B runs and 129 the nine MomaGraph-R1 runs. Because the observed times include contention, this figure overstates what the same runs cost on dedicated hardware, and we report it as the resource we consumed rather than as a throughput benchmark. Evaluating a final checkpoint over the 800 test questions takes a few minutes; the 2,000-question validation split and the zero-shot panel add roughly 15 hours in total. Development runs preceding the reported study are not included in this figure, and we do not claim it as the cost of reproducing the work from scratch; it is the cost of the runs reported here.

Reproduction cost. A reader reproducing the headline result needs the two MomaGraph-R1 configurations A and B at both seeds, approximately 58 GPU-hours at the rates above, plus the scene-graph cache described below. The remaining configurations support the interface comparison and the null result, and the zero-shot panel needs inference only.

F.5 Determinism and variance

Runs are seeded but not made bit-deterministic: `-deterministic` is off, so CUDA kernel selection is free to vary. Two runs of the same configuration at the same seed are therefore not guaranteed identical. The variation we report across seeds (for example 71.50 and 71.38 for MomaGraph-R1's text configuration) includes this source alongside initialisation and data-order effects, and we do not attempt to separate them. The main intervals quantify episode-sampling variation conditional on fitted checkpoints, excluding training-seed uncertainty. Table 11 uses a question bootstrap.

G Human label audit

G.1 Artifact scope and sensitivity

Evidence-strip provenance. The reference rules for Presence-change and Global-count are defined over the sampled evidence strip, but the strip indices are *not* persisted in the released episode records. The evidence a label refers to therefore cannot be reconstructed from the release by a model, an annotator, or a reader. This is a defect of the artifact rather than annotation noise, and persisting one field per episode would make every label in these families checkable. We retain both families in all reported results and verify in Section 5 that removing them (222 of the 800 test questions) preserves the text-over-control direction, with gains of $+5.80$ points on MomaGraph-R1 and $+3.46$ on Gemma-3-4B on the remaining 578, both intervals excluding zero.

We audited the label quality of the two question families whose reference rules are most exposed to a mismatch between what the question asks and what the rule computes: Presence-change and Global-count. This appendix reports the protocol, results, wording-versus-rule differences, and implications for interpreting the benchmark.

G.2 Selection of the audited families

Presence-change carries the largest single family effect we report (+23.5 points on MomaGraph-R1), compared with -0.5 on Gemma-3-4B; an options-only probe reaches 49.0% on this family. No arm beats chance on Global-count, and the memory effect there is -3.8 points. Auditing both families tests the reference rules behind a large gain and a concentration of residual errors.

G.3 Protocol

Why the whole episode rather than the cited frames. Our first annotation build showed only each question's cited evidence frames, which provided insufficient context for episode-spanning questions. One Global-count item asks how many distinct decoration objects the robot saw *anywhere*

Sample	25 questions per family, fixed seed, interleaved so that presentation order carries no information about family or answer.
Evidence shown	*Every* frame of the episode, as a contact sheet, with each question's cited evidence frames additionally listed as text.
Reference answer	Hidden from annotators.
Response options	The question's own options, plus an explicit *"cannot tell, even from the whole trajectory"*.
Annotators	Four, working independently from the same file. One is an author of this paper; the reference answers were hidden from all four. One annotator did not complete every item, so per-annotator denominators differ.
Tooling	A self-contained offline HTML annotation file; scoring by a separate script.

in the house while citing two frames, both in the same bathroom. An episode-spanning question cannot be settled from a two-frame subset, so every Global-count item would have returned "cannot tell" as an artifact of the interface. The model does not answer from the cited frames either. It answers from a memory summary accumulated over the whole trajectory, so showing annotators the whole episode is the comparison that matches what the model is given. Section G.7 explains the cost of that choice.

G.4 Results

Table 9: Human accuracy against the reference answer, four independent annotators. Chance is computed from each family's actual option counts: Presence-change is binary; the sampled Global-count items comprise 16 four-option and 9 three-option questions. Intervals are bootstrap 95% intervals over items. Parenthesised counts are "cannot tell" responses, which are excluded from the accuracy but reported because they measure how often the evidence was judged insufficient.

Family	A1	A2	A3	A4	Chance
Presence-change	62.5% [42, 79]	56.5% [35, 78]	66.7% [48, 86]	63.6% [45, 82]	50.0%
cannot tell	(1/25)	(2/25)	(4/25)	(1/23)	
Global-count	38.1% [19, 57]	17.4% [4, 35]	17.6% [0, 35]	44.4% [22, 67]	28.0%
cannot tell	(4/25)	(2/25)	(8/25)	(0/18)	

Agreement with reference answers by family. On Presence-change all four annotators score above the 50% binary chance level, between 56.5% and 66.7%. No single interval excludes chance at this sample size, but four readers out of four pointing the same way suggests they do extract something from the episode. On Global-count they do not: accuracies span 17.4% to 44.4% around a 28.0% chance level, two of the four fall below it, and one annotator declined 8 of 25 items as unanswerable. The spread across annotators on that family is 27 points, wider than the distance from chance to the best of them.

Inter-annotator agreement. Mean pairwise agreement is 46.5% across the six annotator pairs, ranging from 34.0% to 63.4%. Raw agreement is not comparable across families with different option counts (a binary family agrees half the time by luck), so we also report agreement corrected for the chance agreement implied by each item's own option count:

$$\kappa = \frac{p_o - p_e}{1 - p_e} = \frac{0.465 - 0.397}{1 - 0.397} = 0.114,$$

that is, close to chance. Agreement is lower with four annotators than with two ($\kappa = 0.180$ on the first pass), suggesting that the ambiguity warrants review of question wording, evidence scope, and presentation.

Candidate reference-label disagreements. Table 10 reports where annotator consensus differs from the stored reference.

Majority-vote reference. With four annotators a majority resolves most items: 14 of 50 remain ties and are excluded. The surviving majority accuracies are **70.0% [50, 90]** on Presence-change

Table 10: Items on which annotators agree with each other and the reference answer differs, at two agreement thresholds. We report these as *candidate* label errors, not as an error rate; see Section G.7. Unanimity across four annotators is demanding; the majority row is more informative.

Threshold	Presence-change	Global-count
Unanimous (4/4)	2/23	1/18
Majority (3/4)	6/25	5/25

over 20 items, against 50.0% chance, and **31.2% [12, 56]** on Global-count over 16 items, against 28.0% chance. These descriptive human baselines have different point estimates, while both intervals include their respective chance levels.

G.5 Question wording and reference-rule scope

The reference explanations reveal a specific source of disagreement: the rule and the question wording can refer to different observation scopes. A representative Presence-change explanation reads:

"Painting is still present at the strip's final position 5—it never disappeared."

Twelve of the 25 sampled Presence-change references use that construction. The rule encodes **terminal absence over a six-frame sampled strip**, while the question asks whether an object *"was visible at some point but then gone from view later"*. These scopes differ; the disagreements are consistent with readers interpreting the broader question wording.

Global-count references show the same pattern from the other direction: they are computed *"across 2 rooms of the strip"* while the question says *"anywhere in the house"*.

Recovering the sampled evidence strip. The sampled strip determines both reference rules, but the released episode records contain no field identifying which frames the strip comprised. We confirmed this against the episode schema. Reconstructing the evidence used by each rule therefore requires the missing strip indices. Persisting this field would permit direct comparison of reference answers and annotations under the same evidence scope.

G.6 Sensitivity of the reported conclusions

We retain both families in all headline results and check that their removal preserves the text-over-control direction. Dropping Presence-change and Global-count entirely removes 222 of the 800 test questions, leaving 578.

Table 11: Memory effects excluding both audited families. Question-bootstrap 95% intervals over the 578 questions that remain after removing Presence-change and Global-count. MomaGraph-R1 seed 1's $B-A$ and seed 0's $B-C$ both read $+5.54$ [$+2.2$, $+8.8$]; this is a coincidence confirmed by independently recomputing each from its own per-question records, not a duplicated cell.

	$B-A$, all 800	$B-A$, 578	$B-C$, 578
MomaGraph-R1 seed 0	$+7.25$	$+6.06$ [$+2.8$, $+9.5$]	$+5.54$ [$+2.2$, $+8.8$]
MomaGraph-R1 seed 1	$+6.00$	$+5.54$ [$+2.2$, $+8.8$]	$+6.40$ [$+2.9$, $+9.9$]
Gemma-3-4B seed 0	$+3.50$	$+3.63$ [$+0.2$, $+7.1$]	$+3.29$ [-0.2, $+6.7$]
Gemma-3-4B seed 1	$+2.50$	$+3.29$ [$+0.2$, $+6.4$]	$+3.11$ [$+0.0$, $+6.2$]

The route-history dissociation survives the same test: dropping Presence-change alone, the effect on the remaining route-history questions is $+16.40$ [$+9.39$, $+23.48$] on MomaGraph-R1 and $+12.70$ [$+6.22$, $+19.15$] on Gemma-3-4B, both intervals excluding zero on $n = 189$.

Reduced-set interface comparison. Gemma's $B - C$ contrast is borderline on the reduced set: $+3.29$ [-0.2, $+6.7$] at seed 0 and $+3.11$ [$+0.0$, $+6.2$] at seed 1, one interval just including zero and the other just excluding it. MomaGraph-R1 holds at both seeds. The reduced-set interface comparison supports MomaGraph-R1; Gemma remains directionally consistent.

G.7 Interpretation of the audit

Candidate disagreements. Our annotators saw every frame of the episode, while the reference rules are defined over the sampled strip. This evidence mismatch can increase disagreement, particularly for Presence-change: more frames give more opportunities to observe an object leave view. We therefore report Table 10 as candidate disagreements and do not convert it into an error rate. Additional annotators would help characterize agreement, while persisting the strip indices is needed to align the evidence shown with the reference rule. The audit identifies the wording-versus-rule mismatch documented in Section G.5, which the reference explanations state themselves.

Interface-specific human performance. Scanning 8–43 thumbnails for a small object is a different task from reading a structured memory summary. These figures bound what the *questions* support given this interface; they are not a clean measurement of human ability on the underlying task.

Small samples. After removing "cannot tell" responses, 21–24 items remain per family per annotator, hence intervals roughly 35 points wide.

Direction of counting disagreements. We expected humans to systematically *undercount* on Global-count, since the visibility log contains objects a viewer cannot readily pick out of a thumbnail. They do not: across items, annotator answers were lower than the reference 5 times, equal 8 times, and higher 7 times. These counts show no consistent undercounting pattern and motivate closer examination of the question scope.

G.8 Model and human performance under different evidence interfaces

Table 12: Accuracy on the two audited families. Two human references are given: the best single annotator of the four, and the majority vote over them, which is the more defensible baseline because it is not selected post hoc.

	Presence-change	Global-count
Human, best of four	66.7%	44.4%
Human, majority vote	70.0%	31.2%
Arm A, no memory	58.3%	38.8%
Arm B, memory text	**81.9%**	35.0%
Arm D, graph + text	66.7%	36.2%

Global-count is difficult for both models and annotators; the human majority vote (31.2%) has an interval that includes chance. On Presence-change arm B exceeds the majority vote by 11.9 points and the best single annotator by 15.2, and we cannot currently separate two readings of that gap.

(a) Suitability of the memory representation. "Visible, then gone" is near-mechanical to read off first- and last-seen frame stamps, whereas a human must scan up to 43 thumbnails for a small object. This is consistent with the paper's thesis.

(b) Alignment with the reference rule. Presence-change references are derived symbolically from the per-frame visible-object log, and the memory text is rendered from a *predicted* scene graph over those same frames. The model therefore reads a noisy version of the source the reference was computed from, while the human reads pixels. Predicted graphs carry 46.8% out-of-vocabulary object mentions. They remain coupled to the reference through the shared frames, while annotators interpret the pixels directly.

Distinguishing these explanations requires matched evidence interfaces. The sensitivity check in Section G.6 shows that the text-over-control direction persists when both audited families are excluded.

G.9 REPRODUCTION

The annotation file, the four response files, the sampling manifest, and the scoring script are included in the code release. The manifest contains the reference answers and is the one file that must be withheld from annotators.

```
python scripts/analysis/score_label_audit.py \
    logs/analysis/human/label_audit_manifest.json \
    logs/analysis/human/label_audit_*.json
```

The scorer accepts any number of annotator files and reports per-annotator accuracy, pairwise and chance-corrected agreement, majority vote, and candidate errors under unanimity and majority.

H STATISTICAL ESTIMATORS AND COMPLETE RUN RESULTS

For episode e, question j, and matched training seed s, let c^B_{esj} and c^A_{esj} be binary correctness. The seed-averaged paired difference is

$$d_{ej} = \frac{1}{|S|}\sum_{s\in S}(c^B_{esj} - c^A_{esj}), \qquad \widehat{\Delta}_Q = \frac{\sum_{e,j}\mathbf{1}\{(e,j)\in Q\}d_{ej}}{\sum_{e,j}\mathbf{1}\{(e,j)\in Q\}}. \tag{21}$$

We sample 200 episode indices with replacement for each of 20,000 bootstrap draws, retaining every selected episode's questions and recomputing numerator and denominator. The 2.5th and 97.5th percentiles give the reported intervals. This procedure also applies to subgroups whose episode-level question counts differ. The random seed is 20260920. Averaging training seeds before resampling prevents two evaluations of the same question from being counted as two independent questions.

These intervals quantify test-episode sampling variation conditional on the fitted checkpoints. They do not include training-seed uncertainty, label error, model-selection effects, or uncertainty in split construction. Family intervals are pointwise descriptive intervals, not multiplicity-adjusted discoveries. The graph-only near-zero contrasts are not equivalence tests. A practical equivalence margin and adequate power would have to be specified for that claim.

House dependence and development history. The production manifest is required to count distinct test houses and determine whether multiple episodes share a house. Episode resampling does not account for that additional dependence; if houses repeat, house-clustered uncertainty is the appropriate next check. The intervals should not be read as estimates of generalization to independently sampled houses without this verification. Reference development notes explicitly describe inspecting a 200-episode test split to revise the time axis, vocabulary canonicalization, and truncation policy of memory text. Those historical notes leave the exact question version unspecified. The development process therefore includes test-set feedback, with uncertain overlap with the current labels. Using final checkpoints avoids selecting each run by its best observed validation score, but does not remove adaptive representation design. A fresh sealed evaluation would be needed for confirmatory evidence.

The raw MomaGraph A/B/D means are 64.8125%, 71.4375%, and 68.6250%, respectively. The B–A contrast is 6.625 percentage points. Reporting a mean of rounded seed percentages can shift the final displayed digit; all new tables are generated directly from integer correctness records.

The untuned comparison contains 387/800 correct without memory and 333/800 with memory. Its paired difference is -6.75 points. The difference between the two-seed trained effect and this untuned effect is 13.375 points. This arithmetic contrast is descriptive, not a causal estimate of a uniquely identified memory-reading mechanism. The available exports do not fully archive the untuned launch invocation.

Table 13: Complete final supervised correctness counts. Every reported configuration has two seeds on both backbones; the collapsed run and the untuned comparison are separated below. Validation uses 2,000 questions and test 800; the online training probes use a smaller subset and are not these scores. Arm C is reported at w=2000, the w=400 run having failed to train (Section J). The untuned rows are not fine-tuned and so have no validation score.

Backbone	Arm / seed	Test count	Test (%)	Validation count
MomaGraph-R1	A, seed 0	514/800	64.25	1326/2000
MomaGraph-R1	A, seed 1	523/800	65.38	1325/2000
MomaGraph-R1	B, seed 0	572/800	71.50	1449/2000
MomaGraph-R1	B, seed 1	571/800	71.38	1406/2000
MomaGraph-R1	C, seed 0	513/800	64.13	1342/2000
MomaGraph-R1	C, seed 1	510/800	63.75	1348/2000
MomaGraph-R1	D, seed 0	548/800	68.50	1377/2000
MomaGraph-R1	D, seed 1	550/800	68.75	1374/2000
Gemma-3-4B	A, seed 0	505/800	63.13	1330/2000
Gemma-3-4B	A, seed 1	514/800	64.25	1333/2000
Gemma-3-4B	B, seed 0	533/800	66.63	1364/2000
Gemma-3-4B	B, seed 1	534/800	66.75	1339/2000
Gemma-3-4B	C, seed 0	505/800	63.13	1358/2000
Gemma-3-4B	C, seed 1	517/800	64.63	1330/2000
Gemma-3-4B	D, seed 0	525/800	65.63	1361/2000
Gemma-3-4B	D, seed 1	535/800	66.88	1323/2000
MomaGraph-R1	C , seed 0	265/800	33.13	713/2000
Untuned MomaGraph	No memory	387/800	48.38	–
Untuned MomaGraph	Text	333/800	41.63	–

Table 14: All matched test contrasts in percentage points. Unqualified contrasts average both seeds before episode resampling; rows marked s0 and s1 give the per-seed values. MomaGraph arm C is the w=2000 configuration. Intervals are conditional on these fitted models.

Backbone	Contrast	Difference [95% CI]
MomaGraph-R1	B–A	+6.62 [+3.94, +9.38]
MomaGraph-R1	B–A, s0	+7.25 [+4.12, +10.38]
MomaGraph-R1	B–A, s1	+6.00 [+3.12, +8.88]
MomaGraph-R1	D–A	+3.81 [+1.62, +6.00]
MomaGraph-R1	D–A, s0	+4.25 [+1.38, +7.12]
MomaGraph-R1	D–A, s1	+3.38 [+0.62, +6.12]
MomaGraph-R1	B–D	+2.81 [+0.62, +5.12]
MomaGraph-R1	B–D, s0	+3.00 [+0.12, +5.88]
MomaGraph-R1	B–D, s1	+2.62 [-0.38, +5.62]
MomaGraph-R1	C–A	-0.88 [-2.25, +0.50]
MomaGraph-R1	C–A, s0	-0.12 [-2.12, +2.00]
MomaGraph-R1	C–A, s1	-1.62 [-3.62, +0.25]
Gemma-3-4B	B–A	+3.00 [+0.69, +5.31]
Gemma-3-4B	B–A, s0	+3.50 [+0.62, +6.38]
Gemma-3-4B	B–A, s1	+2.50 [-0.25, +5.25]
Gemma-3-4B	D–A	+2.56 [+0.44, +4.75]
Gemma-3-4B	D–A, s0	+2.50 [-0.12, +5.12]
Gemma-3-4B	D–A, s1	+2.62 [-0.12, +5.50]
Gemma-3-4B	B–D	+0.44 [-1.00, +1.81]
Gemma-3-4B	B–D, s0	+1.00 [-1.12, +3.12]
Gemma-3-4B	B–D, s1	-0.12 [-2.38, +2.25]
Gemma-3-4B	C–A	+0.19 [-0.94, +1.31]
Gemma-3-4B	C–A, s0	0.00 [-1.75, +1.75]
Gemma-3-4B	C–A, s1	+0.38 [-1.12, +2.00]
Untuned MomaGraph	Text–no memory	-6.75 [-11.00, -2.50]

I QUESTION-FAMILY RESULTS AND SENSITIVITY ANALYSES

I.1 ROUTE LENGTH AND RECORD BUDGETS

History and route length are different.

The largest text-memory gains occur in the same-room S1 condition, 11.75 points for MomaGraph-R1 and 8.00 for Gemma-3-4B. The gains on the longest route stratum, S4, are 5.75 and 3.75 points respectively. Remembering earlier sightings can matter within one room; crossing more rooms does not produce a monotonic increase in the observed benefit. The intermediate S2/S3 gains are 6.00/3.00 points for MomaGraph and 1.25/−1.00 for Gemma. All contrasts average both seeds; each stratum contains 50 episodes and 200 questions. These patterns do not isolate timestamp or topology effects.

Removing spatial relations has little measured effect.

We re-score the text configuration with the record re-rendered at nine budgets, varying the per-room object cap from four to unlimited and the retained-relation cap from zero to unlimited. Every budget-contrast interval includes zero, and the whole grid spans 2.00 points. Deleting every *spatial relation costs 0.75 points [−2.50, +1.00], and its effect on the route-history families is −2.06 [−5.50, +1.03]. The result is consistent with a benefit from room-grouped object lists and frame stamps, whose contributions are not separately isolated. This is an intervention on the record rather than an inference from which families improve, and it agrees with the family-level result. The intervals leave uncertainty about smaller effects; we do not claim the record is saturated at the shipped budget.*

Figure 2 summarizes the text-memory contrasts and Table 2 gives the per-family accuracies behind them. The tables below retain all arms and matched family comparisons. The route-history grouping reproduces the overall gain through a weighted average; this is an accounting identity, not an independent validation of a mechanism. Removing Presence-change preserves the text-over-

Table 15: MomaGraph-R1 matched family-level effects in percentage points. All three contrasts average both seeds before episode resampling; arm C is the w=2000 configuration. Intervals are pointwise and exploratory, and are not adjusted for testing eight families.

Family	B–A	C–A	D–A
Order	+17.12 [+5.48, +28.77]	0.00 [-6.85, +6.85]	+8.22 [-0.68, +17.12]
Presence-change	+23.53 [+12.75, +34.80]	-0.49 [-2.45, +1.47]	+8.33 [+1.96, +14.71]
Rooms-visited	+15.95 [+6.47, +25.43]	-3.45 [-9.05, +2.16]	+16.38 [+7.33, +25.43]
Counting	-3.08 [-8.46, +2.31]	-1.54 [-6.15, +3.08]	-0.77 [-6.15, +3.85]
Existence	0.00 [-4.42, +4.42]	+2.21 [-0.88, +5.31]	+1.33 [-1.77, +4.42]
Global-count	-3.75 [-8.75, +1.25]	-4.58 [-8.75, -0.83]	-2.50 [-9.17, +4.17]
Persistence	+2.67 [-1.91, +7.52]	+1.15 [+0.00, +2.67]	-1.53 [-5.00, +1.91]
Room Identification	+1.25 [-2.50, +5.62]	0.00 [-4.38, +4.38]	+1.25 [-2.50, +5.62]

Table 16: Gemma-3-4B matched family-level effects in percentage points. All three contrasts average both seeds before episode resampling. Intervals are pointwise and exploratory, and are not adjusted for testing eight families.

Family	B–A	C–A	D–A
Order	+4.11 [-6.16, +14.38]	0.00 [-6.85, +6.85]	+3.42 [-4.79, +11.64]
Presence-change	-0.49 [-6.86, +6.37]	-0.98 [-4.41, +2.45]	-0.98 [-6.86, +4.90]
Rooms-visited	+18.10 [+9.48, +26.72]	-2.59 [-6.03, +0.43]	+16.38 [+6.90, +25.86]
Counting	+0.77 [-5.38, +6.92]	+1.54 [-2.31, +5.38]	-0.77 [-6.15, +4.62]
Existence	+2.21 [-1.77, +6.19]	+0.88 [-2.21, +3.98]	+2.21 [-1.77, +6.64]
Global-count	+3.75 [-2.50, +10.00]	+0.83 [-2.50, +4.17]	+1.25 [-4.17, +6.67]
Persistence	-3.44 [-6.92, +0.00]	+1.53 [+0.00, +3.46]	-3.44 [-6.54, -0.38]
Room Identification	-3.12 [-6.88, +0.62]	+0.62 [-2.50, +3.75]	+1.25 [-3.12, +5.62]

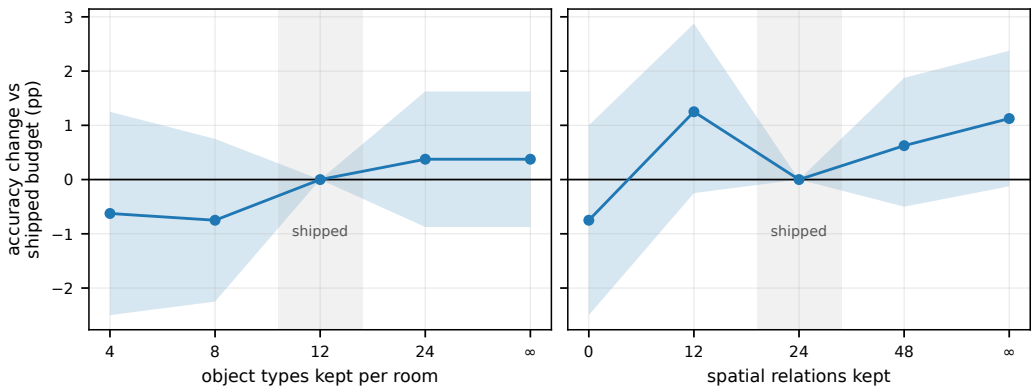


Figure 6: **Sensitivity to text-record budgets and spatial relations.** Arm B re-scored with the record re-rendered at nine budgets, as change against the shipped setting with paired bootstrap 95% bands. Every band crosses zero and the whole grid spans 2.00 points; deleting *every* spatial relation costs 0.75 points, $[-2.50, +1.00]$.

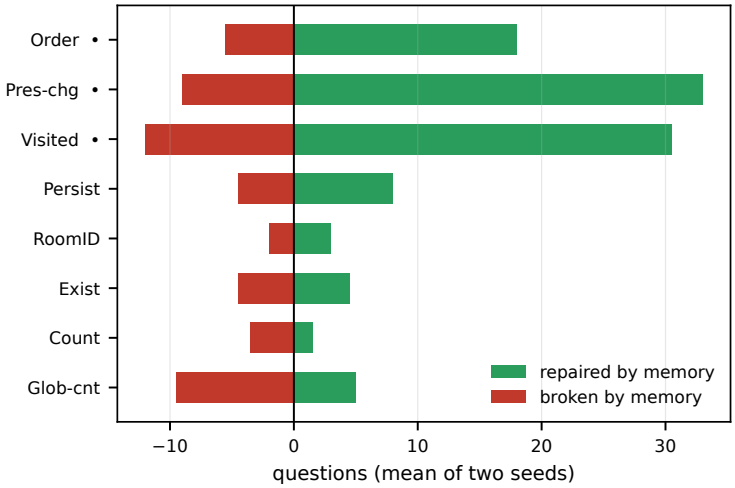


Figure 7: **Where memory repairs answers and where it breaks them.** Questions the no-memory control answered incorrectly that text memory fixes (green) and correct answers it breaks (red), averaged over MomaGraph-R1's two seeds; route-history families marked •. Memory repairs 104 and breaks 50, a net 53 of 800, recovering the $+6.63$ point effect from answer changes. Displayed repair and regression counts are rounded; the net uses unrounded means.

no-memory improvement in both backbones. On MomaGraph, the text-over-combined difference becomes smaller and its interval includes zero.

Rooms-visited can include set-based room counts, so removing it tests sensitivity to a family that may be answerable without visit order. The current correctness exports support this family-level analysis. Determining the operation required by each question additionally requires its complete production record. The route-stratum contrasts do not show a monotonic increase in memory benefit with longer routes. Same-room episodes can still require retention across viewpoints, and question-family composition can differ by stratum. Without matched family-by-horizon controls and the full route/evidence manifest, we do not assign a causal effect to route length.

Table 17: Sensitivity to removing Presence-change. The full remaining set has 698 questions. Order and Rooms-visited together have 189 questions. All contrasts average both seeds and no new model is trained for this analysis.

Backbone	Contrast	All remaining	Order + Rooms-visited
MomaGraph-R1	B–A	+4.15 [+1.48, +6.91]	+16.40 [+9.39, +23.48]
MomaGraph-R1	D–A	+3.15 [+0.80, +5.50]	+13.23 [+6.70, +19.79]
MomaGraph-R1	B–D	+1.00 [-1.08, +3.13]	+3.17 [-1.09, +7.54]
Gemma-3-4B	B–A	+3.51 [+1.14, +5.90]	+12.70 [+6.25, +19.15]
Gemma-3-4B	D–A	+3.08 [+0.71, +5.51]	+11.38 [+5.16, +17.66]
Gemma-3-4B	B–D	+0.43 [-1.13, +1.97]	+1.32 [-2.22, +4.95]

Table 18: Sensitivity excluding Rooms-visited. Order + Presence-change has 175 questions across 151 episodes. The entire remaining test set has 684 questions. All contrasts average both seeds. The grouped gain in Gemma is not statistically resolved after removing Rooms-visited.

Backbone	Contrast	Order + Presence-change	All except Rooms-visited
MomaGraph-R1	B–A	+20.86 [+12.87, +29.10]	+5.04 [+2.36, +7.73]
MomaGraph-R1	D–A	+8.29 [+2.91, +13.53]	+1.68 [-0.44, +3.78]
MomaGraph-R1	B–D	+12.57 [+6.25, +18.98]	+3.36 [+0.94, +5.79]
MomaGraph-R1	C–A	-0.29 [-3.41, +2.89]	-0.44 [-1.75, +0.87]
Gemma-3-4B	B–A	+1.43 [-4.55, +7.40]	+0.44 [-1.79, +2.65]
Gemma-3-4B	D–A	+0.86 [-4.00, +5.75]	+0.22 [-1.75, +2.19]
Gemma-3-4B	B–D	+0.57 [-4.19, +5.17]	+0.22 [-1.46, +1.88]
Gemma-3-4B	C–A	-0.57 [-3.89, +2.84]	+0.66 [-0.51, +1.89]

Table 19: Exploratory supervised effects by route stratum. Episode IDs join the final correctness exports to route metadata; point estimates are cross-checked against evaluator stratum summaries. Every stratum has 50 episodes and 200 questions. Intervals resample episodes and are conditional on the fitted models.All contrasts average both seeds.

Backbone	Contrast	Stratum	Effect [95% CI]
MomaGraph-R1	B–A	S1	+11.75 [+5.36, +18.18]
MomaGraph-R1	B–A	S2	+6.00 [+0.73, +11.18]
MomaGraph-R1	B–A	S3	+3.00 [-2.44, +8.65]
MomaGraph-R1	B–A	S4	+5.75 [+1.20, +10.28]
MomaGraph-R1	B–D	S1	+2.75 [-1.60, +7.35]
MomaGraph-R1	B–D	S2	+2.50 [-2.25, +7.03]
MomaGraph-R1	B–D	S3	+3.25 [-1.82, +8.41]
MomaGraph-R1	B–D	S4	+2.75 [-0.88, +6.49]
Gemma-3-4B	B–A	S1	+8.00 [+3.50, +12.75]
Gemma-3-4B	B–A	S2	+1.25 [-3.00, +5.50]
Gemma-3-4B	B–A	S3	-1.00 [-5.50, +3.50]
Gemma-3-4B	B–A	S4	+3.75 [-0.50, +8.00]
Gemma-3-4B	B–D	S1	+0.75 [-1.75, +3.00]
Gemma-3-4B	B–D	S2	-0.25 [-3.00, +2.25]
Gemma-3-4B	B–D	S3	+0.25 [-2.75, +3.25]
Gemma-3-4B	B–D	S4	+1.00 [-2.25, +4.25]
MomaGraph-R1	C–A	S1	-0.75 [-3.75, +2.50]
MomaGraph-R1	C–A	S2	-0.50 [-3.00, +2.00]
MomaGraph-R1	C–A	S3	-1.00 [-4.00, +1.75]
MomaGraph-R1	C–A	S4	-1.25 [-4.00, +1.25]
Gemma-3-4B	C–A	S1	-0.75 [-2.75, +1.25]
Gemma-3-4B	C–A	S2	+1.25 [-1.25, +3.75]
Gemma-3-4B	C–A	S3	-0.75 [-3.25, +1.75]
Gemma-3-4B	C–A	S4	+1.00 [-1.25, +3.00]

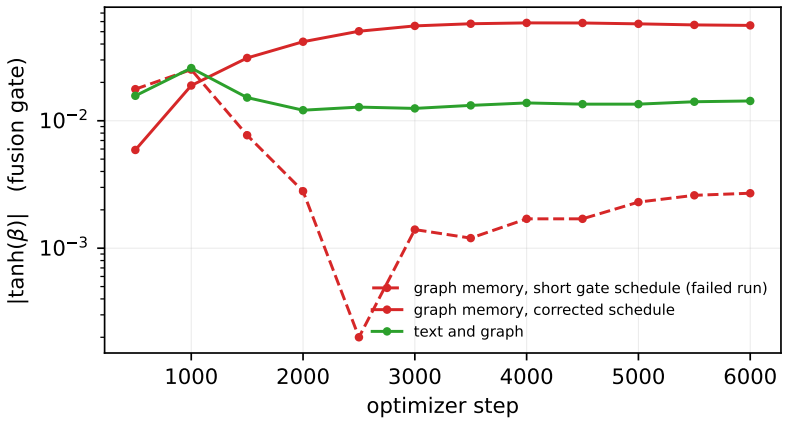


Figure 8: **Recovering from an observed graph-only collapse.** Magnitude of the learned output-fusion gate $|\tanh(\beta)|$ over training, log scale. Under the shorter gate schedule the gate rises briefly, falls to 2×10^{-4} before partially recovering. Changing the schedule length alone gives a gate that rises monotonically to 0.056, twenty times the failed run's final value. *The recovered run still does not beat its no-memory control*, and reproduces at a second seed.

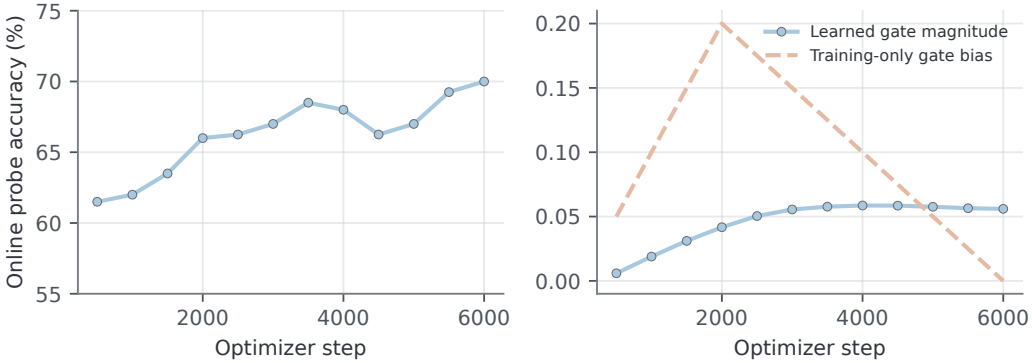


Figure 9: Recovered graph-only run with gate warmup $w = 2000$. Left: the online answer probe during training, distinct from full validation and test evaluation. Right: learned output-gate magnitude and the training-only scaffold. The scaffold reaches zero only at the final step. The final test result is 513/800 correct.

J Graph optimization and correctness diagnostics

The completed short-warmup graph-only run produces 265/800 correct, against 261/800 for a constant-A answer rule. Its dominant-letter share is approximately 94%. This diagnostic has one verified final evaluation; the seed-1 run terminates at step 100. Table 13 lists the final results used here.

The recovered graph-only run has 513/800 correct; the seed-matched no-memory control has 514/800. At the item level, their correctness differs on 65 questions: 33 favor the control and 32 favor graph memory. On Gemma, graph-only and no-memory each score 505/800, with 25 correct-to-incorrect changes and 25 incorrect-to-correct changes. This rules out the claim that equality of aggregate accuracy means the model made no changes. It does not identify their cause.

The available final exports record correctness rather than selected answer letters. Consequently, the preceding matrices report agreement in correctness. Two models can agree that an item is wrong while selecting different wrong options. Four-arm consensus and the per-item oracle describe the

Table 20: Seed-0 correctness patterns across A/B/C/D, using recovered MomaGraph C. The per-item oracle selects a correct arm after seeing the label and is not an executable policy. The reported agreement concerns correctness; selected answer letters may differ.

Backbone	All correct	All wrong	Mixed	Oracle (%)
MomaGraph-R1	427	152	221	81.00
Gemma-3-4B	417	192	191	76.00

Table 21: MomaGraph-R1 pairwise agreement in correctness (%), seed 0. Two different wrong letters count as agreement here because exports store correctness rather than letters.

	A	B	C	D
A	100.00	80.50	91.88	83.25
B	80.50	100.00	81.13	85.25
C	91.88	81.13	100.00	84.63
D	83.25	85.25	84.63	100.00

diversity of outcomes on this set; achieving the oracle score would additionally require a selection rule evaluated without access to the labels.

The archived flip analyses also compare the sets on which C differs from A and D differs from B. Overlap above an independence reference is evidence that some items are sensitive to graph access in both configurations. It is not evidence that the changed answers are more grounded. The exact interpretation is further limited by C/D graph filtering differences and the longer warmup of recovered MomaGraph C.

J.1 The gate warmup schedule: what it controls and what it does not

The training-only scaffold $b(s)$ rises from 0 to 0.2 over w optimizer steps and decays to 0 over the next $2w$, so it vanishes at step $3w$: step 1,200 for $w = 400$ and step 6,000, the final step, for $w = 2000$. Only the two arms with a graph pathway are affected; A and B have no fusion gate.

Three things follow, and the third is the one that matters as shown in Table 23.

The schedule controls convergence. One completed run collapsed: MomaGraph's graph-only arm under the short schedule, whose gate fell to 2×10^{-4} and whose answers became 94% one letter. Lengthening w alone raised the final gate from 0.0027 to 0.0560, roughly twentyfold, and the test result from 265 to 513 correct. The same short schedule did not collapse Gemma's graph-only arm, which scores 505/800 against a control of 505/800 (Table 13), so the failure is specific to a backbone and arm rather than to w itself.

It does not control usefulness. The recovered run has the widest gate we observe, 0.0560, and scores 513/800 against a seed-matched no-memory control of 514/800; at the second seed, 0.0544 and 510/800 against 523/800. The graph-only arm never exceeds its control at either schedule or either seed on either backbone. Gate magnitude and accuracy are unrelated across this range: opening the pathway changed whether it trained, not whether it helped.

The combined arm's gate is modest, and its schedule was not searched. Every text-and-graph run ends between 0.0078 and 0.0143, several times above the collapsed run, but four to seven times

Table 22: Gemma-3-4B pairwise agreement in correctness (%), seed 0. Two different wrong letters count as agreement here because exports store correctness rather than letters.

	A	B	C	D
A	100.00	82.25	93.75	85.25
B	82.25	100.00	82.00	90.00
C	93.75	82.00	100.00	85.25
D	85.25	90.00	85.25	100.00

Table 23: Learned output-fusion gate by arm, schedule and seed, with the seed-matched test result and its no-memory control. Minimum and final are of $|\tanh(\beta)|$ over the recorded validation points, read at the last optimizer step; the training-only scaffold $b(s)$ is zero at inference throughout. Arm D also receives the rendered text, so its margin over the no-memory control is not attributable to the graph pathway. Gate traces were not retained for Gemma's seed-1 runs or for its graph-only run beyond step 1,000, so those rows are omitted; their test results appear in Table 13.

Backbone	Arm	w	Seed	Min	Final	Test	No-mem. control
MomaGraph-R1	C	400	0	0.0002	0.0027	265/800	514/800
MomaGraph-R1	C	2000	0	0.0059	0.0560	513/800	514/800
MomaGraph-R1	C	2000	1	0.0060	0.0544	510/800	523/800
MomaGraph-R1	D	400	0	0.0121	0.0143	548/800	514/800
MomaGraph-R1	D	400	1	0.0040	0.0079	550/800	523/800
Gemma-3-4B	D	400	0	0.0070	0.0078	525/800	505/800

below the recovered one. These arms also carry the rendered text, so their margin over the no-memory control is not a graph effect; text alone scores higher still. We did not train the combined arm at $w = 2000$, leaving that cell of the design untested. Because opening the gate twentyfold in the graph-only arm produced no gain over its control, we do not attribute the text-over-combined difference in Section 5.3 to the schedule, though we cannot rule it out. A matched schedule sweep over both graph arms is the experiment this analysis calls for.

K Validity probes, model panel, and evidence acquisition

K.1 Evidence ladder

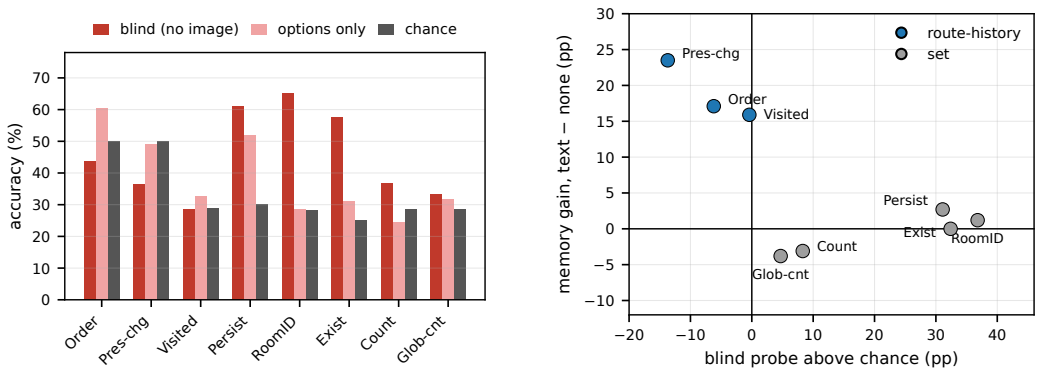
Table 24: **Evidence access and acquisition on Qwen3-VL-4B.** These zero-shot protocols are separate from the supervised comparison. Accuracy includes unparsed responses as errors; coverage is the fraction parsed. Oracle replay succeeds on 198 usable trajectories from 200 attempts.

Evidence condition	Correct / 800	Accuracy (%)	Coverage (%)
Options only	312	39.00	100.00
Question + options + target room	363	45.38	99.50
Closed-loop exploration	364	45.50	100.00
Referenced oracle frames	468	58.50	100.00
Oracle replay	459	57.38	99.00

Table 25: Reported Qwen3-VL-4B (Qwen Team, 2025b) family probe results (%). Counts underlying the percentages are recovered from the reported family summaries, not independently rescored against a complete current gold split. Six chance baselines are available in evaluator metadata; the evaluator omits the last two families.

Family	n	Blind	Options only	Chance
Room Identification	80	65.00	28.75	28.23
Persistence	131	61.07	51.91	29.96
Existence	113	57.52	30.97	25.07
Order	73	43.84	60.27	50.00
Counting	65	36.92	24.62	28.59
Presence-change	102	36.27	49.02	50.00
Global-count	120	33.33	31.67	–
Rooms-visited	116	28.45	32.76	–

The six available family-specific chance baselines differ materially from the overall 33.0% level. Order and Presence-change are binary in the recorded family summaries, with 50% chance. Order's blind accuracy is below that baseline; its options-only accuracy is above it. Counting's blind accuracy exceeds its own approximately 28.6% chance, despite being near the aggregate benchmark chance. We therefore avoid screening families against a single global threshold.



(a) Image-free probes against per-family chance.

(b) Probe advantage against memory gain.

Figure 10: **What each family leaks, and where memory pays.** Image-free probes on Qwen3-VL-4B against each family's own chance level (a): Room Identification is answerable 37 points above chance with no image at all, whereas Order and Presence-change both sit *below* chance blind. That organises b, where the two groups separate on both axes: every route-history family scores at or below chance blind and gains at least 15.9 points from text memory, while every set family scores above chance blind and gains at most 2.7. Question-weighted, the blind probe is 6.5 points below chance on the route-history group and 23.2 points above it on the set group. Memory pays where a language prior fails, and pays nothing where the prior already succeeds. This association does not rule out option cues or identify shortcut use.

The previous audit singled out Room Identification, Persistence, Existence, and Order, covering 397 questions, based on their reported blind or options-only performance. That grouping is descriptive; no formal statistical screening threshold is documented. It is neither a per-question shortcut certificate nor an exhaustive list of families above their own chance levels. The correctness exports alone do not recover current probe gold labels. Family probe counts shown here are reconstructed from reported family percentages and denominators, with that provenance stated in the caption.

Table 26: Full-population and parsed-only answer accuracy. Every condition designates 800 questions. Reporting the parsed-only column without coverage changes the evaluation population.

System	Correct	Parsed	All (%)	Parsed only (%)
Oracle frames	468	800	58.50	58.50
Closed loop	364	800	45.50	45.50
Oracle replay	459	792	57.38	57.95
MomaGraph closed loop	54	128	6.75	42.19

Table 27: Qwen3-VL-4B (Qwen Team, 2025b) evidence-ladder accuracy by route stratum (%). Each stratum has 50 episodes and 200 questions. Supervised contrasts obtained by joining episode IDs to this route metadata appear in Table 19.

Evidence	S1	S2	S3	S4
Oracle frames	61.50	53.50	61.00	58.00
Closed loop	49.50	44.50	50.00	38.00
Oracle replay	61.50	52.50	60.00	55.50

Image-versus-blind differences across the zero-shot panel (Table 2) can combine visual information with changes in answer parsing. Qwen2.5-VL-32B (Team, 2025) parses 620/800 blind responses but all 800 image responses. LLaVA (Liu et al., 2024) parses 733/800 image responses and all 800 blind responses. A raw accuracy difference is therefore not a clean visual-grounding effect. MomaGraph's panel adapter also consumes only the last referenced image, while several other panel systems receive all referenced images. This difference prevents a matched multi-image ranking.

The Qwen3 (Qwen Team, 2025b) evidence ladder has a consistent backbone but deliberately changes evidence access. Oracle frames and oracle replay are information references. Closed-loop exploration must acquire the evidence itself. The replay record contains 198 usable navigation trajectories; two attempted episodes lack usable replay navigation outputs. We report 198/198 conditional success and 198/200 attempted-run success separately. The MomaGraph closed-loop condition uses a Done-gated answer-invocation policy and is separate from this matched-backbone ladder (Appendix L).

L Error anatomy: repairs, shared errors, and execution

Table 28: **What text memory repairs.** Each seed answers 800 questions; both backbones have two seeds. Repairs are A-wrong/B-correct; regressions are A-correct/B-wrong. Net change is $100(\text{repairs} - \text{regressions})/800$ points. Count shading uses a fixed 0–800 scale.

Backbone	Seed	Repairs	Regressions	Net (pp)
MomaGraph-R1	0	107	49	+7.25
MomaGraph-R1	1	100	52	+6.00
Gemma-3-4B	0	85	57	+3.50
Gemma-3-4B	1	74	54	+2.50

We analyze the existing 200-episode, 800-question predictions, separating supervised constrained-choice outcomes from generation-based evidence conditions. No additional model runs are used. Transitions measure changes in correctness without identifying their cause.

L.1 Accuracy as a balance of repairs and regressions

For source configuration X and target Y, let $c_i^X, c_i^Y \in \{0,1\}$ be correctness on question i. Define

$$R^+_{X\to Y} = \sum_{i=1}^{N} (1 - c_i^X) c_i^Y, \qquad R^-_{X\to Y} = \sum_{i=1}^{N} c_i^X (1 - c_i^Y), \tag{22}$$

$$E_Y = \sum_{i=1}^{N} (1 - c_i^Y), \qquad \mathrm{Acc}_Y - \mathrm{Acc}_X = \frac{R^+_{X\to Y} - R^-_{X\to Y}}{N}. \tag{23}$$

Here $N = 800$ for each seed. The equality is exact: repairs and regressions account for the reported accuracy difference. We retain seed-specific counts, then average differences as in Appendix H. Repeated predictions on the same question are not additional independent test questions. Blue shading on transition and coverage counts uses a common 0–800 scale; percentage cells retain the 0–100 scale. MomaGraph B repairs 37.4% and 36.1% of the errors made by its corresponding A runs, while changing 9.5% and 9.9% of previously correct answers to incorrect ones. Gemma's

Table 29: Complete correctness transitions for selected supervised comparisons. The first configuration is the source and the second the target. Each row partitions 800 questions. C denotes recovered MomaGraph or completed Gemma graph-only.

Backbone	Source–target	Seed	Both correct	Both wrong	Repair	Regression
MomaGraph-R1	A–B	0	465	179	107	49
MomaGraph-R1	A–B	1	471	177	100	52
Gemma-3-4B	A–B	0	448	210	85	57
Gemma-3-4B	A–B	1	460	212	74	54
MomaGraph-R1	B–D	0	501	181	47	71
MomaGraph-R1	B–D	1	496	175	54	75
Gemma-3-4B	B–D	0	489	231	36	44
Gemma-3-4B	B–D	1	487	218	48	47
MomaGraph-R1	A–C	0	481	254	32	33
Gemma-3-4B	A–C	0	480	270	25	25
Gemma-3-4B	A–C	1	491	260	26	23

Table 30: Conditional rates for A to B. Repair rate divides by A errors; regression rate divides by A correct answers. The distinct denominators are shown explicitly. These percentages are descriptive.

Backbone	Seed	Repair rate (%)	Regression rate (%)	A errors	A correct
MomaGraph-R1	0	37.41	9.53	286	514
MomaGraph-R1	1	36.10	9.94	277	523
Gemma-3-4B	0	28.81	11.29	295	505
Gemma-3-4B	1	25.87	10.51	286	514

corresponding rates are 28.8% and 25.9% repaired, against 11.3% and 10.5% regressed. Because denominators differ, counts determine the net gain; conditional rates cannot be subtracted.

L.2 Which questions are repaired?

The route-history group contributes 82 repairs against 28 regressions in MomaGraph seed 0 and 81 against 25 in seed 1. The other 509 questions contribute 25 against 21 and 19 against 27. Gemma contributes 58 against 28 and 43 against 26 in the route-history group, and 27 against 29 and 31 against 28 elsewhere. This decomposition locates the net improvement in the same families as the aggregate analysis. Across MomaGraph seeds, 72 questions are repaired in both runs and 21

Table 31: MomaGraph-R1, seed 0: A-to-B transitions by family. Net change is a within-family percentage-point difference. Stable errors are questions wrong in both configurations.

Family	n	Repair	Regression	Net (pp)	Stable errors
Counting	65	2	3	-1.54	22
Existence	113	5	4	+0.88	11
Global-count	120	6	8	-1.67	70
Order	73	17	5	+16.44	13
Persistence	131	8	4	+3.05	19
Presence-change	102	33	8	+24.51	10
Room Identification	80	4	2	+2.50	10
Rooms-visited	116	32	15	+14.66	24

Table 32: MomaGraph-R1, seed 1: A-to-B transitions by family. Net change is a within-family percentage-point difference. Stable errors are questions wrong in both configurations.

Family	n	Repair	Regression	Net (pp)	Stable errors
Counting	65	1	4	-4.62	19
Existence	113	4	5	-0.88	11
Global-count	120	4	11	-5.83	67
Order	73	19	6	+17.81	18
Persistence	131	8	5	+2.29	16
Presence-change	102	33	10	+22.55	9
Room Identification	80	2	2	0.00	11
Rooms-visited	116	29	9	+17.24	26

regress in both. The shared repairs include 27 Presence-change, 22 Rooms-visited, and 12 Order questions. These three families account for 61 of 72 repeated repairs. The overlap shows that many improvements recur on particular items in the two completed seeds. It does not estimate their frequency under an unrestricted population of training seeds. At episode level, B improves the number of correct answers on 78 and 75 MomaGraph episodes, compared with 34 and 39 episodes with fewer correct answers. Gemma improves 59 and 56 episodes and worsens 39 and 40. An episode can contain both transitions, so equal scores can conceal different question outcomes.

L.3 A shared concentration in global counting

Global-count has the highest text-memory error rate in every run: 78/120 for both MomaGraph seeds, and 72/120 and 80/120 for Gemma. It accounts for 78/228, 78/229, 72/267 and 80/266 of

Table 33: Gemma-3-4B, seed 0: A-to-B transitions by family. Net change is a within-family percentage-point difference. Stable errors are questions wrong in both configurations.

Family	n	Repair	Regression	Net (pp)	Stable errors
Counting	65	3	6	-4.62	20
Existence	113	7	4	+2.65	12
Global-count	120	13	6	+5.83	66
Order	73	15	8	+9.59	21
Persistence	131	1	8	-5.34	25
Presence-change	102	9	9	0.00	29
Room Identification	80	3	5	-2.50	12
Rooms-visited	116	34	11	+19.83	25

Table 34: Gemma-3-4B, seed 1: A-to-B transitions by family. Net change is a within-family percentage-point difference. Stable errors are questions wrong in both configurations.

Family	n	Repair	Regression	Net (pp)	Stable errors
Counting	65	5	1	+6.15	22
Existence	113	5	3	+1.77	9
Global-count	120	16	14	+1.67	66
Order	73	10	11	-1.37	24
Persistence	131	5	7	-1.53	19
Presence-change	102	7	8	-0.98	33
Room Identification	80	0	3	-3.75	14
Rooms-visited	116	26	7	+16.38	25

Table 35: Episode-level A-to-B changes in the number of correctly answered questions. Improved, tied, and worsened sum to 200. The final column separately counts episodes containing at least one repair and one regression; it overlaps the first three columns.

Backbone	Seed	Improved	Tied	Worsened	Both changes
MomaGraph-R1	0	78	88	34	12
MomaGraph-R1	1	75	86	39	12
Gemma-3-4B	0	59	102	39	15
Gemma-3-4B	1	56	104	40	11

Table 36: Residual text-memory errors by family. Each model cell is error rate in percent (error count); shading encodes the percentage, with darker cells indicating more errors. The last column counts questions wrong in all four text-memory runs.

Family	n	Moma s0	Moma s1	Gemma s0	Gemma s1	Shared errors
Counting	65	38.46 (25)	35.38 (23)	40.00 (26)	35.38 (23)	13
Existence	113	13.27 (15)	14.16 (16)	14.16 (16)	10.62 (12)	8
Global-count	120	65.00 (78)	65.00 (78)	60.00 (72)	66.67 (80)	55
Order	73	24.66 (18)	32.88 (24)	39.73 (29)	47.95 (35)	10
Persistence	131	17.56 (23)	16.03 (21)	25.19 (33)	19.85 (26)	17
Presence-change	102	17.65 (18)	18.63 (19)	37.25 (38)	40.20 (41)	6
Room Identification	80	15.00 (12)	16.25 (13)	21.25 (17)	21.25 (17)	9
Rooms-visited	116	33.62 (39)	30.17 (35)	31.03 (36)	27.59 (32)	24

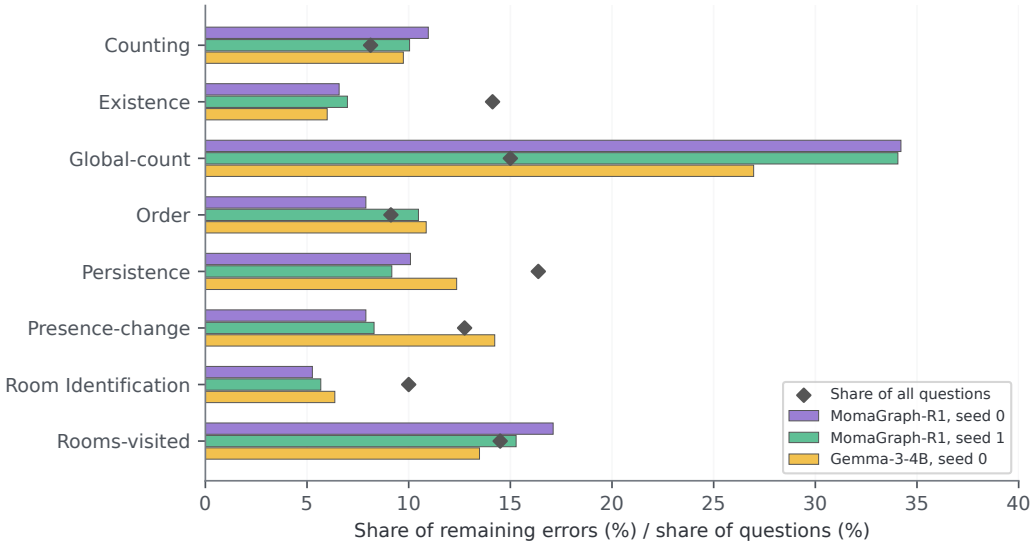


Figure 11: Composition of the remaining text-memory errors. Each bar is a family's share of a model's total errors; dark diamonds show its share of the 800 questions. Global-count contributes a larger share of errors than its question share in each of the three runs plotted here (MomaGraph-R1 seeds 0–1 and Gemma-3-4B seed 0); Gemma-3-4B seed 1 is not shown but follows the same pattern (Table 36). Shares use each run's own error denominator, not a pooled sample.

their remaining errors. Across all four B runs, 142 questions remain incorrect; 55 are Global-count. Across all 15 completed noncollapsed supervised runs, 93 questions remain incorrect, including 39 Global-count questions. These shared errors identify questions for label review and targeted memory improvements.

The architecture offers a concrete hypothesis for further work: type-level maximum simultaneous counts do not preserve persistent instance identity (Equation 15). The observed concentration makes an instance-aware counting comparison relevant. Correctness alone does not locate errors in perception, aggregation, interpretation, or readout.

L.4 Graph access redistributes correctness

Adding graph access to text repairs 47 and 54 B errors in MomaGraph, while regressing 71 and 75 B-correct answers. Gemma repairs 36 and 48, and regresses 44 and 47. The three-way comparison with A makes this redistribution more specific. D retains 63/107 and 55/100 of MomaGraph B's repairs, and 57/85 and 55/74 of Gemma B's. D also answers 21, 21, 12 and 23 questions, respectively, that both A and B miss. Thus, D has lower aggregate accuracy but some complementary answers.

Compared with no memory, graph-only access produces 32 repairs and 33 regressions for recovered MomaGraph C, and 25 of each and 26 against 23 for Gemma C. Near-equal aggregate scores therefore coexist with changed correctness. These results describe configurations with the filtering and optimization settings in Table 6; they do not isolate graph topology or a particular injection pathway. The sensitivity intervals excluding Presence-change remain the appropriate qualification for the B–D comparison.

L.5 Separate acquiring evidence, producing an answer, and scoring it

For a scheduled test question, let Z_i indicate a recorded parsed answer. With unparsed or unattempted answers scored as incorrect, the empirical identity is

$$\widehat{\mathrm{Acc}}_{\mathrm{all}} = \frac{N_{\mathrm{parsed}}}{N_{\mathrm{test}}} \widehat{\mathrm{Acc}}_{\mathrm{parsed}}, \qquad N_{\mathrm{parsed}} = \sum_i Z_i. \tag{24}$$

Table 37: Coverage decomposition for generation-based evidence conditions. Each row partitions 800 questions. Null raw records and absent prediction rows are distinguished from nonempty responses without a parsed answer. Generation differs from supervised scoring.

Condition	Correct	Parsed wrong	Null raw	Unparsed text	No row
Qwen oracle frames	468	332	0	0	0
Qwen closed loop	364	436	0	0	0
Moma closed loop	54	74	672	0	0
Qwen oracle replay	459	333	0	0	8

Here $N_{\text{test}} = 800$ includes questions without an answer invocation or prediction record. A parsed record can arise through permissive extraction, so $Z_i = 1$ is not itself a guarantee that the response explicitly chose an option. The following cases show why the execution policy and raw output matter alongside this arithmetic.

Case 1: an ordinary article becomes an option letter. In the Qwen3-VL-4B blind probe, question `ep_0108_q2` produces a response beginning:

> None of these items are typically found in a bedroom, and the question asks what the robot saw in the bedroom — but the context is a bathroom.

The saved response ends mid-phrase, "However, if we must choose the *least incorrect* or *", yet its record contains `parsed=A` and `parse_ok=true`. Applying the reference parser to this exact string reproduces A: explicit-answer patterns find no choice, and a case-insensitive standalone-letter fallback matches the ordinary article "a". This is a verified extraction artifact. It explains the conversion of a non-choice response into a letter; correctness still depends on the reference answer. Supervised candidate scoring does not use this fallback.

Case 2: a room-context objection accompanies a non-answer. A second blind-probe response, `ep_0113_q1`, states:

> The robot is in the Living Room, and the question asks what it saw in the Bedroom — which is not the same location.

It ends "we cannot logically select" and remains unparsed. The reference probe supplies a target-room label together with the question and choices. The response treats the stated room context as evidence about the robot's location and declines to choose. The quoted text is the model's response, not a reconstruction of the original question or an independently verified room conflict.

Case 3: termination controls whether QA is invoked. MomaGraph closed-loop episode `ep_0001` records eight actions, zero traveled distance, zero invalid actions, and `stuck` termination. Its four QA records contain null raw responses. The reference execution policy invokes QA only after Done. Across the run, all 168 stuck episodes have four null responses each; the other 32 episodes contribute all 128 parsed answers. Of these, 54 are correct. The 16% coverage is therefore tied to answer invocation, rather than 672 observed formatting errors. Navigation action counts and termination summaries do not identify why the policy stalled.

Cases 4–5: navigation success and answer success diverge. Qwen3-VL-4B closed-loop episode `ep_0011` ends with `target_not_visible` after one action and zero travel, yet all four questions are scored correct. By contrast, oracle-replay episode `ep_0003` reaches its target after 32 actions and 11.75 m, with SPL 0.9367, but answers only two of four questions correctly. These are two different episodes and evidence protocols, not a matched causal comparison. Together they illustrate why successful navigation, usable observations, and correct answers should be reported separately. The first case leaves the contribution of initial observations and priors unresolved; the second evaluates supplied-trajectory replay.

Case 6: a repair and a regression repeated across seeds. Within the Order family, MomaGraph question `ep_0008_q4` is incorrect in both A seeds and correct in both B seeds. Question

ep_0003_q1 exhibits the reverse pattern. Recovered C seed 0 follows A on both items; D is correct on the first only in seed 0 and wrong on the second in both seeds. These correctness-only examples show heterogeneous outcomes within one family. We attach no reconstructed question, generated explanation, or temporal-reversal diagnosis to them.

What this analysis contributes. The supervised records identify which questions improve or regress between configurations. The generation records distinguish parsing errors, skipped answer calls, and evidence acquisition. Together, these records support retaining paired per-question outcomes, auditing raw-answer parsing, reporting evaluated and answered denominators, and separating navigation success from QA accuracy.

M PROMPT CATALOG

The following listings document the committed reference prompt templates and substitutions. Their version relationship to the deployed protocol follows Appendix D. Unicode punctuation is normalized for typesetting; the extraction artifact retains the original strings.

M.1 PER-FRAME SCENE-GRAPH CONSTRUCTION

The task_instruction slot is contextual. The output must describe the current image. In the reference implementation, the first question supplies this context. The distinction between current room and target room must survive parsing.

scene_graph.txt

```
You are an embodied robot perception system. Construct a scene graph of ONLY the CURRENT
↪  image from visible evidence. Building the scene graph is the task -- do NOT answer the
↪  question.

Task instruction (context only -- do NOT answer it): {task_instruction}

Reason first inside <SCENE_GRAPH_ANALYSIS>...</SCENE_GRAPH_ANALYSIS>, then emit exactly
↪  one JSON object inside <SCENE_GRAPH_JSON>...</SCENE_GRAPH_JSON>.

STAGE 1 -- ROOMS (judge from the IMAGE, never from the instruction text):
- scene_type: the room you are CURRENTLY IN, inferred from the visible objects/layout.
↪  Exactly one of: "Bathroom", "Bedroom", "Kitchen", "LivingRoom".
- target_scene: the room implied by the task instruction (where the target object lives).
↪  Same 4 values, or "unknown". This is the ONLY place the room named in the instruction
↪  may go -- it must NEVER set scene_type.

STAGE 2 -- OBJECTS (nodes): every object ACTUALLY VISIBLE now (furniture, appliances,
↪  fixtures, doors, windows, decorations, and navigation structures like
↪  door/hallway/entrance). Number duplicates (chair1, chair2). Do NOT hallucinate hidden
↪  objects. If scene_type != target_scene, the target object is NOT in this room -- do
↪  NOT invent it; list only the visible navigation structures (door/hallway/entrance)
↪  that could lead out.

STAGE 3 -- EDGES: relationships between visible objects only, using exactly these values:
- spatial_relationships: left_of, right_of, in_front_of, behind, higher_than, lower_than,
↪  close, far, touching
- functional_relationship: openorclose, adjust, control, providepower, activate, pairwith,
↪  support
- action_type: navigate, press, rotate, pull, push, open, close, insert, pick, release
- function_type: navigation, open_close_control, device_control, power_supply,
↪  water_flow_control, parameter_adjustment, special_function, assembly

Output EXACTLY (valid JSON only; do NOT answer the question):
<SCENE_GRAPH_ANALYSIS>
One short paragraph: which room you are in and the visible objects that prove it, whether
↪  the target room differs (so navigation is required), and the key visible
↪  objects/relationships.
</SCENE_GRAPH_ANALYSIS>
<SCENE_GRAPH_JSON>
{{"scene_type": "...", "target_scene": "...", "nodes": [...], "edges": [{{"object1":
↪  "...", "object2": "...", "spatial_relationships": [...], "functional_relationship":
↪  "...", "action_type": "...", "function_type": "...", "is_touching": true}}]}}
</SCENE_GRAPH_JSON>
```

M.2 NAVIGATION

The current scene-graph text and candidate actions accompany a real image. The classification-token insertion is performed by the model wrapper.

navigation.txt

```
### Task: Predict the next navigation action.
### Question: {question}
### Choices: {choices}
### Candidate Actions: (0) MoveAhead (1) RotateLeft (2) RotateRight
↪  (3) LookUp (4) LookDown (5) Done
### Predicted Scene Graph:
{current_predicted_scene_graph}
### Guidance:
Use the scene graph and the question to choose the best next action.
Choose Done only when sufficient information has been gathered to
↪  answer the question.
### Action Token: <CLS>
```

M.3 ANSWERING WITHOUT TEXTUAL MEMORY

The reference wording mentions visual information collected during navigation. Under the reported corrected supervised protocol, the answer image is blank; this wording does not grant access to a stored image sequence.

answer_no_memory.txt

```
### Task: Navigation has finished.
Answer the question using the visual information collected during
↪  navigation.
### Question: {question}
### Choices: {choices}
### Guidance:
Select the correct answer from the given options.
### Answer:
```

M.4 ANSWERING WITH TEXTUAL MEMORY

The memory slot contains the rendered observation record. Arm B bypasses both graph-injection paths; arm D retains them. The question's actual option count determines candidate scoring.

answer_memory.txt

```
### Task: Navigation has finished.
Answer the question using the memory collected during navigation.
### Question: {question}
### Choices: {choices}
### Memory:
{memory_string}
### Guidance:
Select the correct answer from the given options.
### Answer:
```

M.5 QUESTION-GENERATION REFERENCE PROMPT

The released `mem_eqa_v1` questions use symbolic templates over the per-frame visibility log (Appendix C). The reference generator also supports a vision-language question-writing path. We provide both full prompts below to document that optional path and its revisions.

question_generation_base.txt

```
You are an expert annotator for Embodied Question Answering (EQA) datasets.

You are shown a chronological strip of {n_frames} frames captured by a robot exploring a
↪  house. Frames are numbered 0 to {n_frames_minus_1} in the order they occurred, purely
↪  for YOUR bookkeeping. Each frame is labeled with the room the robot was in and a list
↪  of objects visible in that frame, detected by the robot's sensors.

You do NOT know why the robot was moving or where it was heading. Do not assume any frame
↪  is a "destination" or "goal" -- treat every frame as an equally arbitrary snapshot of
↪  the trajectory. Do not use phrases like "the object the robot was looking for" or "the
↪  room the robot needed to reach."

FRAME DATA:
{frame_listing}

Generate EXACTLY {n_questions} multiple-choice questions about this trajectory. Cover a
↪  MIX of:

SINGLE-OBSERVATION (grounded in one frame):
- Existence: was object X visible [in the <room>]?
- Counting: how many X were visible [in the <room>]?
- State: was X open/closed/on/off [in the <room>]?
- Spatial relation: what was X next to / left of / above, from the robot's point of view?

CROSS-OBSERVATION (require comparing multiple frames -- prioritize these, at least half
↪  your questions):
- Room identification: in which room did the robot see X? (answer is a room name)
- Order: did the robot see X before or after Y?
- Presence-change: was X visible at some point but gone later?
- Comparison: was there more of category X in one room than another?
- Persistence: across how many of the rooms visited was X visible?

CRITICAL -- NEVER write "frame N" (or any numeral) inside the question text itself. The
↪  frame numbers above exist only so YOU can track which frames ground which question; a
↪  robot answering these later has no notion of frame indices, only its own lived
↪  trajectory. Refer to timing and location the way a person would: by room name, by
↪  relative order ("before/after", "first/later"), or not at all when a frame doesn't
↪  need distinguishing.

OTHER RULES:
- Every question must be answerable ONLY from the frame data given above -- do not invent
↪  objects or rooms not listed.
- Do not phrase any question as "Is the {{object}} in the {{room}} ...", i.e. never state a
↪  specific object-room pairing as a given fact in the question setup -- an object-room
↪  pairing may be the ANSWER to a question (e.g. room identification) but must never be
↪  handed to the solver as the premise.
- Keep questions concise and unambiguous. Use plain object type names (e.g. "chair", not
↪  "the ArmChair_17").
- For every question, set "frame_refs" to the list of frame INDEX NUMBERS (from the
↪  0..{n_frames_minus_1} labeling above) that the answer depends on -- this is a data
↪  field for our records, separate from the question text. Single-observation questions
↪  have exactly one. Cross-observation questions must have two or more.
- Spread frame_refs across the available frames -- do not let every question point at the
↪  same frame.

OUTPUT FORMAT:
Return ONLY a valid JSON array with EXACTLY {n_questions} objects and no additional text.

[
  {{
    "question_id": "q_001",
    "category": "Perception",
    "subcategory": "Existence",
    "reasoning_type": "single_frame",
    "difficulty": "Easy",
    "frame_refs": [2],
    "question": "Was a lamp visible at any point in the bedroom?",
    "options": ["A: Yes", "B: No", "C: Partially visible", "D: Cannot determine"],
    "answer": "A",
    "explanation": "Frame 2's object list includes a FloorLamp, room=Bedroom."
  }},
  {{
    "question_id": "q_002",
    "category": "Memory",
    "subcategory": "Room Identification",
    "reasoning_type": "cross_frame",
    "difficulty": "Medium",
    "frame_refs": [1, 4],
    "question": "In which room did the robot see a sofa?",
```

```
    "options": ["A: Bedroom", "B: LivingRoom", "C: Kitchen", "D: It never saw a sofa"],
    "answer": "B",
    "explanation": "A sofa first appears in frame 1's object list, tagged as LivingRoom."
  }},
  {{
    "question_id": "q_003",
    "category": "Memory",
    "subcategory": "Order",
    "reasoning_type": "temporal",
    "difficulty": "Hard",
    "frame_refs": [0, 5],
    "question": "Did the robot see a television before or after it saw a bed?",
    "options": ["A: Before", "B: After", "C: At the same time", "D: It never saw a bed"],
    "answer": "A",
    "explanation": "A television appears in frame 0; a bed first appears in frame 5."
  }}
]
```

M.6 PATCHED QUESTION-GENERATION PROMPT

The patched prompt strengthens frame coverage and answer balance, and specifies the final-frame rule for Presence-change discussed in Appendix G. Both listings are reproduced from the unchanged source files.

question_generation_patched.txt

```
You are an expert annotator for Embodied Question Answering (EQA) datasets.

You are shown a chronological strip of {n_frames} frames captured by a robot exploring a
↪  house. Frames are numbered 0 to {n_frames_minus_1} in the order they occurred, purely
↪  for YOUR bookkeeping. Each frame is labeled with the room the robot was in and a list
↪  of objects visible in that frame, detected by the robot's sensors.

You do NOT know why the robot was moving or where it was heading. Do not assume any frame
↪  is a "destination" or "goal" -- treat every frame as an equally arbitrary snapshot of
↪  the trajectory. Do not use phrases like "the object the robot was looking for" or "the
↪  room the robot needed to reach."

FRAME DATA:
{frame_listing}

Generate EXACTLY {n_questions} multiple-choice questions about this trajectory. Cover a
↪  MIX of:

SINGLE-OBSERVATION (grounded in one frame):
- Existence: was object X visible [in the <room>]?
- Counting: how many X were visible [in the <room>]?
- State: was X open/closed/on/off [in the <room>]?
- Spatial relation: what was X next to / left of / above, from the robot's point of view?

CROSS-OBSERVATION (require comparing multiple frames -- prioritize these, at least half
↪  your questions):
- Room identification: in which room did the robot see X? (answer is a room name)
- Order: did the robot see X before or after Y?
- Presence-change: was X visible at some point but gone later?
- Comparison: was there more of category X in one room than another?
- Persistence: across how many of the rooms visited was X visible?

CRITICAL -- NEVER write "frame N" (or any numeral) inside the question text itself. The
↪  frame numbers above exist only so YOU can track which frames ground which question; a
↪  robot answering these later has no notion of frame indices, only its own lived
↪  trajectory. Refer to timing and location the way a person would: by room name, by
↪  relative order ("before/after", "first/later"), or not at all when a frame doesn't
↪  need distinguishing.

SPREAD YOUR ATTENTION ACROSS THE WHOLE STRIP, NOT JUST THE LAST FRAME. The final frame is
↪  where the robot happened to stop, not the "important" one -- every frame is an equally
↪  arbitrary snapshot (see above). If you notice yourself writing several questions about
↪  the same frame, or the same one or two objects, deliberately pick a different frame
↪  and different objects for the rest of your batch instead.

ANSWER MIX -- do not let every question in a subcategory share the same correct answer.
↪  Concretely:
- Existence: do not only confirm things you saw. For roughly half of your Existence
↪  questions, name an object type that appears in a DIFFERENT frame/room of this same
↪  strip but is genuinely absent from the room you're asking about, so the correct answer
↪  is "No" -- never invent an object type that appears nowhere in the frame data at all.
```

```
- Order: decide which object to name first in the sentence independently of which one the robot actually saw first. Across your batch, some Order questions should correctly resolve to "Before" and some to "After" -- do not always narrate events in the order they happened.
- Presence-change: this is checkable, not a guess -- find the object's LAST appearance across the whole strip, then look specifically at the last-numbered frame ({n_frames_minus_1}). If the object IS in that last frame's object list, the correct answer is "No" (it never disappeared). If the object appeared earlier but is NOT in that last frame's object list, the correct answer is "Yes". Do every Presence-change question this way -- both answers are fine, but only write "No" when you can actually point at the object sitting in frame {n_frames_minus_1}'s list.
If you catch yourself about to give two questions in the same subcategory the same answer, change the object or the phrasing so they differ.

OTHER RULES:
- Every question must be answerable ONLY from the frame data given above -- do not invent objects or rooms not listed.
- Do not phrase any question as "Is the {{object}} in the {{room}} ...", i.e. never state a specific object-room pairing as a given fact in the question setup -- an object-room pairing may be the ANSWER to a question (e.g. room identification) but must never be handed to the solver as the premise.
- Keep questions concise and unambiguous. Use plain object type names (e.g. "chair", not "the ArmChair_17").
- For every question, set "frame_refs" to the list of frame INDEX NUMBERS (from the 0..{n_frames_minus_1} labeling above) that the answer depends on -- this is a data field for our records, separate from the question text. Single-observation questions have exactly one. Cross-observation questions must have two or more.
- Spread frame_refs across the available frames -- do not let every question point at the same frame.

OUTPUT FORMAT:
Return ONLY a valid JSON array with EXACTLY {n_questions} objects and no additional text.

[
  {{
    "question_id": "q_001",
    "category": "Perception",
    "subcategory": "Existence",
    "reasoning_type": "single_frame",
    "difficulty": "Easy",
    "frame_refs": [2],
    "question": "Was a toilet visible at any point in the bedroom?",
    "options": ["A: Yes", "B: No", "C: Partially visible", "D: Cannot determine"],
    "answer": "B",
    "explanation": "Frame 2's object list (room=Bedroom) has no Toilet -- a Toilet only appears in frame 0, tagged room=Bathroom, so it never appeared in the bedroom."
  }},
  {{
    "question_id": "q_002",
    "category": "Memory",
    "subcategory": "Room Identification",
    "reasoning_type": "cross_frame",
    "difficulty": "Medium",
    "frame_refs": [1, 4],
    "question": "In which room did the robot see a sofa?",
    "options": ["A: Kitchen", "B: Bedroom", "C: LivingRoom", "D: It never saw a sofa"],
    "answer": "C",
    "explanation": "A sofa first appears in frame 1's object list, tagged as LivingRoom."
  }},
  {{
    "question_id": "q_003",
    "category": "Memory",
    "subcategory": "Order",
    "reasoning_type": "temporal",
    "difficulty": "Hard",
    "frame_refs": [0, 5],
    "question": "Did the robot see a bed before or after it saw a television?",
    "options": ["A: Before", "B: After", "C: At the same time", "D: It never saw a bed"],
    "answer": "B",
    "explanation": "A television appears in frame 0; a bed first appears in frame 5 -- the bed was named first in the question but was actually seen AFTER the television, so 'After' is correct."
  }},
  {{
    "question_id": "q_004",
    "category": "Memory",
    "subcategory": "Presence-change",
    "reasoning_type": "cross_frame",
    "difficulty": "Medium",
    "frame_refs": [1, 6],
    "question": "Was a bookshelf visible at some point but then gone from view later?",
```

```
    "options": ["A: Cannot determine", "B: Yes", "C: No", "D: Partially visible"],
    "answer": "C",
    "explanation": "A bookshelf appears in frame 1's object list and is still present in
    ↪  frame 6's object list -- it stayed visible, it did not disappear, so the answer is
    ↪  No."
  }}
]
```

N ILLUSTRATIVE EPISODES AND EVALUATED OUTCOME EXAMPLES

N.1 WORKED TEST EXAMPLE

A record the answerer can read.

For test item `ep_0008_q4`, the question asks whether the robot first saw a fridge before or after a houseplant. The retained record places `HousePlant` at frame 6 and `Fridge` at frames 16–22, making "After" the answer. At seed 0, A answers incorrectly while B and D answer correctly. This illustrates the information available to the reader; it does not isolate which component causes the aggregate gain. Figures 12 and 13 show the frames and inputs.

N.2 OBSERVATION-HISTORY EXAMPLE

Table 38 illustrates why retaining earlier sightings can help answer a question after an object leaves view. The summary records observed visibility, not a physical change in the room.

N.3 A VISIBLE DISAPPEARANCE IN THE ILLUSTRATIVE TRAINING COLLECTION

Table 39: Illustrative training questions and source gold labels, separate from the current test set.

ID	Question and options	Gold and family
q1	How many distinct basketball objects did the robot see in the Bedroom? A: 1; B: 4; C: 0; D: 3	A Counting
q2	Was a bed visible at some point but then gone from view later? A: No; B: Yes	B Presence-change
q3	Was a houseplant visible at some point but then gone from view later? A: No; B: Yes	B Presence-change
q4	How many distinct pillow objects did the robot see in the Bedroom? A: 0; B: 4; C: 1; D: 3	C Counting

These four questions are exact source questions, with source option order and reference labels. Counting questions refer to distinct simulator object IDs, whereas the memory renderer retains type-level simultaneous counts. That mismatch illustrates a representational limitation rather than a demonstrated test failure for these examples. The disappearance questions concern leaving the field of view; they do not imply that an object physically vanished or that the simulated world changed.

N.4 MULTI-ROOM OBSERVATION RECORD

A second illustrative training episode visits a living room and a kitchen. Its frames show how multi-room observations accumulate; Table 40 records the source episode and route.

N.5 CURRENT TEST OUTCOMES

The current test examples report item IDs, exported family labels, and correctness under each evaluated configuration. They include both gains and losses to illustrate contrasting outcomes. These

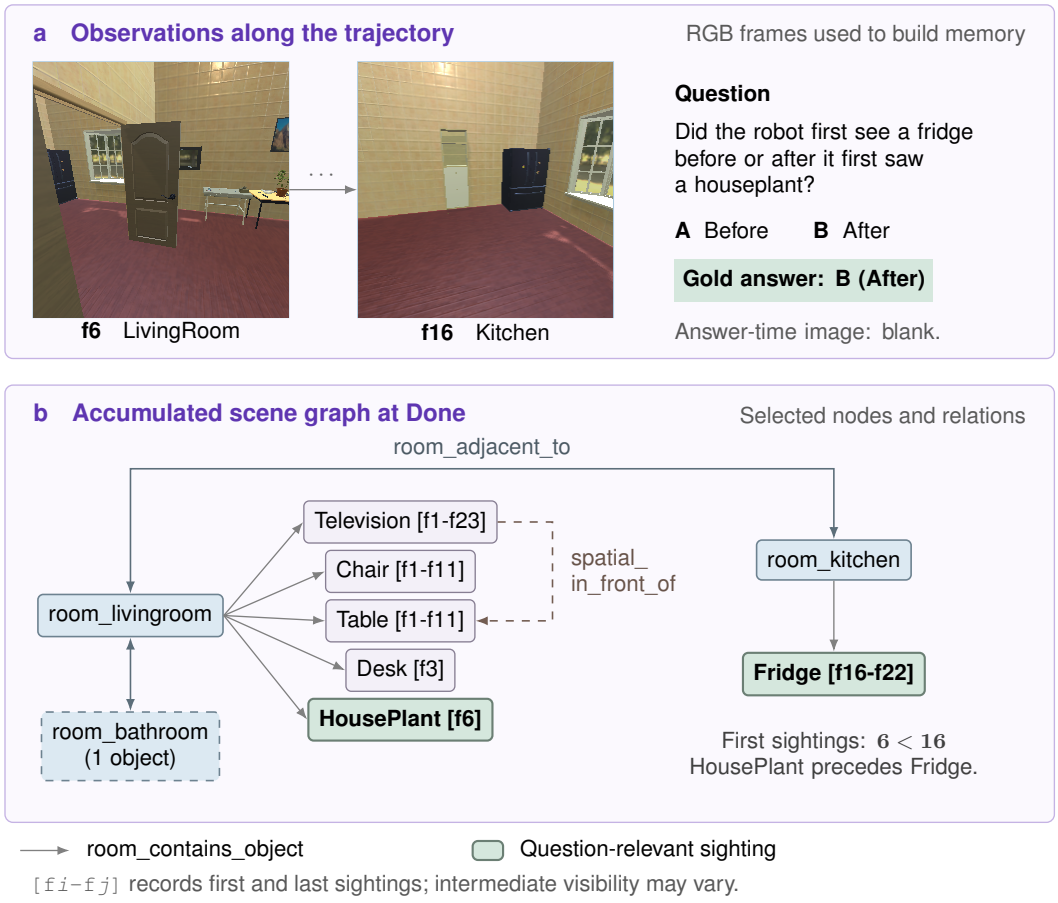


Figure 12: **Trajectory observations and the accumulated scene graph**, test item `ep_0008_q4` (family Order). Panel (a) shows two observations used to construct memory; every arm receives a blank image at answer time. Panel (b) shows selected nodes from the final record at Done, with mint highlighting the question-relevant sightings. `HousePlant` is first seen at frame 6 and `Fridge` at frames 16–22, supporting the answer *After*. Spans report first and last sightings in the final record. This binary example has 50% chance, compared with 33.0% across the dataset; text memory scores 71.2% against a 54.1% no-memory control on Order questions (Table 2).

Table 38: **An observation-history question.** This separate training example illustrates the memory summary using annotation-derived sightings. Frame indices are local; the schematic summary is not a current test prediction.

Observed history	Bed visible in frames 0–1; absent from view in frames 2–5.
Sighting summary	Bedroom; Bed [first = 0, last = 1]; history ends at frame 5.
Source question	Was a bed visible at some point but then gone from view later?
Choices and label	A: No; B: Yes. Annotated answer: B (Yes).

are correctness comparisons rather than reconstructions of model reasoning, and their selection does not estimate population frequencies.

O Development conditions and reproducible analysis

Earlier experiments include smaller splits, graph-wiring diagnostics, altered loss weights or oversampling, and exploratory runs. Their completion does not make them comparable to the final

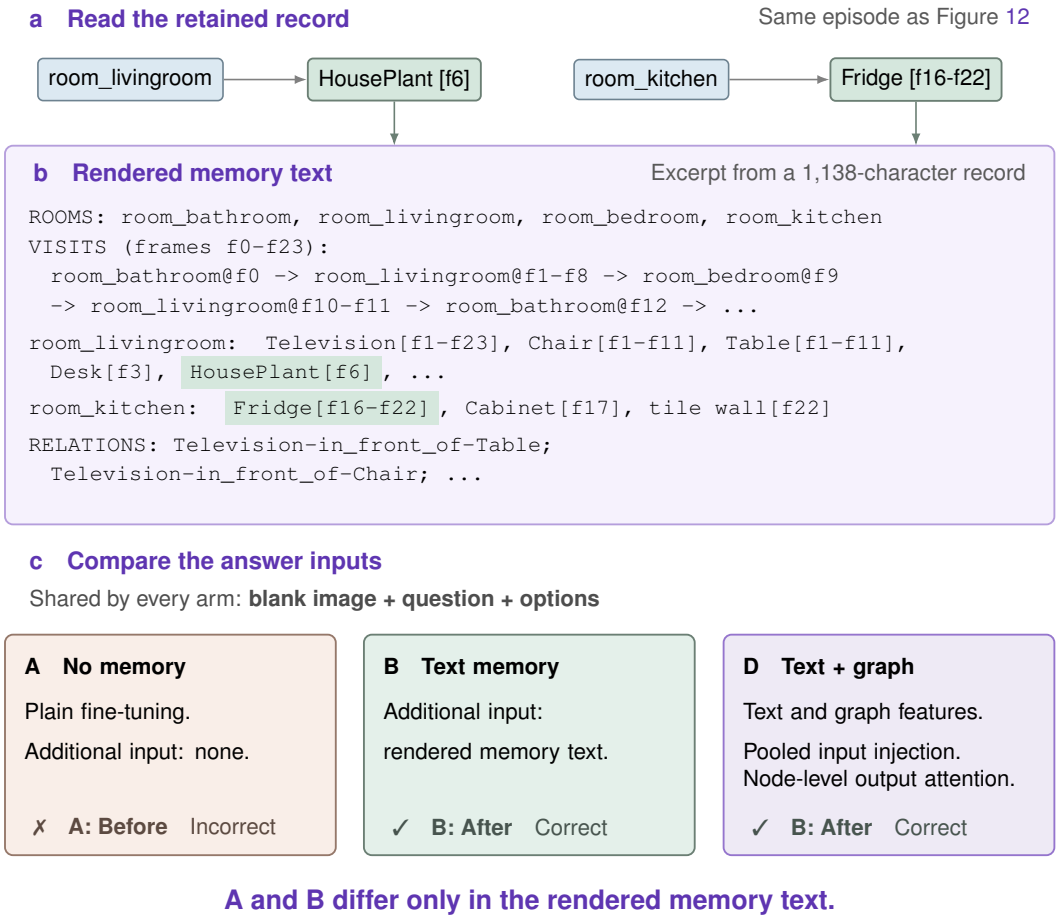


Figure 13: **The rendered memory text and what each arm receives**, same episode. All three arms are given the identical blank answer image; the memory text is the only input that differs between A and B. Arm A (no memory) answers incorrectly; arms B (text) and D (text plus graph features) both answer correctly here. Averaged over the 800-question test set graph access reduces accuracy by 2.81 points relative to text alone (Section 5.3); this example does not isolate that aggregate effect.

Table 40: Living-room-to-kitchen traversal from the separate illustrative training collection.

Field	Source value
Episode	ep_0005
Split	train
House	11
Frames	31
Route	LivingRoom → Kitchen

protocol. The accompanying analysis inventory preserves their configuration labels and recorded status without aggregating their scores into the 200-episode/800-question study.

The key reproducible analysis operations are: load the final correctness exports; assert 800 unique test question IDs and four questions per episode; align arms by ID; average matched-seed differences; resample whole episodes; and retain coverage separately from accuracy. No model weights are loaded for these operations. Historical scripts that bootstrap individual questions or pool two seeds as 1,600 independent observations produce a different uncertainty estimate and are not the source of the intervals reported here.



Figure 14: Frames 0, 2, and 6 of illustrative training episode `ep_0001`, house 2, S1. The initial bedroom view contains a bed and a plant. The middle view still contains the plant but not the bed; the late view shows a painting. This source belongs to a separate training collection, not current test episode `ep_0001`. Its reference questions illustrate the schema and should not be joined to current test predictions by the short episode ID.



Figure 15: Selected early, room-transition, and late views from the second illustrative training episode. These actual source images show the living-room-to-kitchen traversal. No current test response is attached to this illustration.

O.1 Reproduction protocol and extensions

End-to-end reproduction uses the deployed trainer and backbone loaders, episode bundle, split-to-house manifest, perception cache, checkpoints, and evaluation configurations. Question adjudication provides a separate check of visual label validity. The reported metrics concern answer accuracy, navigation, and parsing coverage; computational efficiency is a separate evaluation axis.

Table 41: Selected current-test correctness contrasts, all seed 0. Columns A/B/C/D repeat for MomaGraph then Gemma; MomaGraph C uses recovered warmup. 1 means correct and 0 incorrect. Only correctness and family labels are available for these items.

Item	Family	MomaGraph				Gemma			
		A	B	C	D	A	B	C	D
ep_0079_q4	Counting	1	0	1	1	0	0	0	0
ep_0008_q2	Existence	1	0	1	1	1	0	1	1
ep_0018_q4	Global-count	0	1	0	0	1	1	1	1
ep_0017_q2	Order	1	0	1	1	0	1	0	0
ep_0006_q1	Persistence	0	1	0	0	1	1	1	1
ep_0012_q3	Presence-change	0	1	0	0	0	0	0	0
ep_0123_q3	Room Identification	1	0	1	1	0	1	1	0
ep_0008_q3	Rooms-visited	1	0	1	1	1	1	0	1

Table 42: Treatment of experimental and development conditions. Only protocol-matched final evaluations enter the supervised result tables.

Condition	Purpose	Treatment in this paper
Final A/B/C/D	Supervised memory comparison	Completed final checkpoints; explicit seed counts.
Long graph warmup	Diagnose graph-only collapse	Separate single-seed intervention.
Incomplete graph seeds	Additional replication	No final accuracy imputed.
Untuned answerers	Adaptation context	Separate from trained-arm aggregates.
Real answer images	Earlier evaluation condition	Superseded for the blank-image comparison.
Small-split pilots	Development and debugging	Excluded from current population estimates.
Wiring and loss variants	Optimization diagnostics	No matched final comparison inferred.
Exploration and replay	Evidence acquisition	Separate navigation and coverage denominators.

The present results leave several representation questions open. Matched raw-history text would separate structured rendering from generic extra context. Removing or permuting timestamps would test the temporal annotations. Graph topology perturbation, content-matched node filtering, and isolated injection pathways would test graph-specific explanations. A training interval after warmup-scaffold removal would further test optimization stability. These controls remain untested.

P Extended related work

P.1 Why retain a structured record?

An assistant answering after a traversal needs access to observations that may no longer be in view. A larger context window can hold more evidence, but the observations must still be retained and organized. A scene graph offers one record: object and room nodes, their relations, and annotations of when they were observed. The resulting memory must preserve the information a question needs and expose it in a form the answerer can use.

Our comparison separates this reading problem from autonomous evidence acquisition. All supervised configurations use the same recorded observations and a blank answer image. Text renders the record in the model's language interface; graph features introduce a learned reader. The source is shared, but filtering, budgets, and gate schedules differ, so the experiment compares the implemented configurations rather than an intrinsic advantage of one representation. The related systems below motivate this choice and explain why their end-to-end accuracies are not direct controls.

P.2 Detailed comparison of memory systems

Embodied and episodic question answering. EmbodiedQA studies agents that navigate to answer questions (Das et al., 2018), while OpenEQA distinguishes episodic question answering over an observation history from active evidence acquisition (Majumdar et al., 2024). GraphEQA combines hierarchical scene graphs and image memory for active EQA (Saxena et al., 2025). Extending recall across episodes, Mind Palace links temporally indexed world instances and combines memory with current exploration (Ginting et al., 2025). VL-KnG instead retrieves answers from one persistent knowledge graph built from a tour (Mdfaa et al., 2026). Exploration-aware evaluation separately scores whether an agent looked in the right place from whether it answered correctly (Jiang et al., 2025), and efficient-exploration methods study when an agent should stop gathering evidence before answering (Ren et al., 2024). We focus on the supervised answerer's interface to a record accumulated within one episode, holding answer-time visual access fixed. Appendix Table 43 situates Room to Remember against these and other prior EQA benchmarks on this basis: whether route length is a controlled generation variable, whether multiple memory interfaces are compared on a matched backbone, and whether any condition evaluates with current visual input withheld. VL-KnG instead retrieves answers from one persistent knowledge graph built from a tour (Mdfaa et al., 2026). VSI-

Bench evaluates spatial recall, counting, and appearance order from recorded indoor video without an embodied policy (Yang et al., 2025a).

Learned memory for embodied agents. MomaGraph connects scene-graph representations to embodied reasoning and provides our primary pretrained backbone (Ju et al., 2026). MrSteve organizes what, where, and when information for instruction following in Minecraft (Park et al., 2025). EPM learns updates to textual object-centric state for embodied planning (Parashar et al., 2026). In our study, the text renderer is rule-based and the trained component consumes its output. JanusVLN uses dual implicit spatial and semantic memory for navigation (Zeng et al., 2026), while MemoryVLA studies perceptual-cognitive memory for manipulation (Shi et al., 2026). GraphPad pairs a scene graph with a text log (Ali et al., 2025); VL-MemKnG adds text to a knowledge graph for long trajectories (Lukina et al., 2026). ReMEmbR is the closest prior system to our setting: it builds a long-horizon spatio-temporal memory from a robot's own video and queries it with a language agent to answer navigation questions about what was seen and where (Anwar et al., 2025). It shares our premise that a retained record must be made readable to a language model, and differs in what it measures: ReMEmbR evaluates a retrieval-and-reasoning pipeline end to end, whereas we hold the record fixed and vary only the interface through which one fine-tuned answerer reads it. 3D-Mem stores multi-view memory and frontier snapshots for exploration and spatial reasoning (Yang et al., 2025b), and ConceptGraphs builds open-vocabulary 3D scene graphs from vision-language models for perception and planning (Gu et al., 2024); both construct the kind of structured record we take as given, and neither compares alternative ways of exposing it to the answerer. Event-Grounding Graph unifies spatial and temporal structure from robot observations into one graph (Nguyen et al., 2026), Just-In-Time Scene Graph Growth activates only task-relevant parts of a graph to combat perceptual saturation over long horizons (Chang et al., 2026), and Worth Remembering gates episodic memory formation by surprise within a 4D scene graph (Gorlo et al., 2026). GAVEL extends the same choice to planning, using a graph world model to verify and repair plans (Wang et al., 2026). The variety of successful interfaces motivates comparing text and learned graph access for the same answering task. Our experiments assess the tested configurations, without reproducing or ranking these broader systems.

Memory evaluation and shortcut controls. LongMemEval evaluates persistent conversational memory across distinct capabilities (Wu et al., 2025); REMem studies time-aware graph memory for recollection and reasoning in language agents (Shu et al., 2026). Memory-T1 studies reinforcement learning for temporal reasoning across multi-session agent memory (Du et al., 2026), and World-in-World evaluates world models in a closed loop rather than on static held-out data (Zhang et al., 2026). Both motivate examining which operations a memory supports. In visual grounding, Ref-Adv tests whether apparent reasoning survives shortcut-reducing controls (Dong et al., 2026). For language models reading graphs, Lost in Serialization shows answers can shift with how the same graph is written as text (Herbst et al., 2025). We examine embodied questions with family-level analysis, image-free probes, and separate measurements of answer accuracy and evidence acquisition. These checks help interpret an interface comparison when generated questions can carry exploitable priors.

P.3 Background

Embodied agents built on large models. Vision-language models pretrained on internet-scale data now serve as the default backbone for embodied decision-making, supplying perceptual and language priors that earlier task-specific pipelines lacked (Liang et al., 2025). Because these models still receive only a fixed context of images and text at answer time, work on embodied agents has converged on external structured state – most often a 3D scene graph – as a way to carry information across a trajectory without re-encoding it in full (Rotondi et al., 2026). Knowledge-based question answering over such external structure predates the vision-language-model era (Tan et al., 2023), and interactive question answering in simulated houses was posed even earlier as a way to force an agent to gather its own evidence rather than answer from a single frame (Gordon et al., 2018). Room to Remember inherits this framing: a scene graph is the state, and the open question we study is how a pretrained model should read it.

Why a procedurally generated simulator. Interactive 3D house simulators such as AI2-THOR provide a controllable substitute for real environments, at the cost of relying on a fixed, hand-

Table 43: **Comparison to prior EQA benchmarks. Real**: scenes are captured rather than procedurally generated. **Route**: the number of rooms or door crossings on a route is a directly controlled generation variable, rather than an incidental property of collected scenes. **Interf.**: the benchmark evaluates multiple ways of exposing a retained record to the same pretrained backbone under matched training. **Mem-only**: an evaluation condition withholds current images to isolate memory use.

Benchmark	Environment	Real	Route	Interf.	Mem-only
EmbodiedQA (Das et al., 2018)	House3D	✗	✗	✗	✗
IQA (Gordon et al., 2018)	AI2-THOR	✗	✗	✗	✗
K-EQA (Tan et al., 2023)	AI2-THOR	✗	✗	✗	✗
HM-EQA (Ren et al., 2024)	Habitat/HM3D	✓	✗	✗	✗
OpenEQA (Majumdar et al., 2024)	Habitat/ScanNet	✓	✗	✗	✗
WalkieKnowledge (Mdfaa et al., 2026)	Egocentric video	✓	✗	✗	✗
EXPRESS-Bench (Jiang et al., 2025)	Habitat/HM3D	✓	✗	✗	✗
Room to Remember (Ours)	ProcTHOR/AI2-THOR	✗	✓	✓	✓

authored set of scenes (Kolve et al., 2017). Curated benchmarks including BEHAVIOR-1K instead prioritize simulation fidelity and activity coverage over house-level diversity, evaluating on a fixed catalog of realistic scenes (Li et al., 2024). ProcTHOR takes the opposite trade-off, procedurally sampling floorplans and object placements to generate effectively unlimited, house-disjoint splits from the same asset library (Deitke et al., 2022); this scalability has since motivated procedural generation as a route to training data for physical foundation models more broadly (Lin et al., 2025). We build Room to Remember on ProcTHOR because comparing memory interfaces under house-disjoint train/val/test splits requires scale that no hand-authored scene set provides.

Reading structured state as text or as learned features. A scene graph is only useful to a pretrained vision-language model once it is turned into an input the model can consume, and the field has not converged on one answer. Serializing a graph to natural-language text keeps the model's existing interface intact and has recently been shown to help spatial reasoning when the serialized structure names objects, positions, and attributes explicitly rather than leaving them implicit in pixels (Ma et al., 2026; Fatemi et al., 2024). Encoding the same structure as learned graph features instead asks the model to acquire a new interface during training, echoing the broader graph-versus-token choice surveyed for language models reading graphs generally (Jin et al., 2024; Perozzi et al., 2024). Room to Remember evaluates both choices, and their combination, on the same trained backbone rather than assuming one is preferable.

Table 44: Related study designs. These descriptions do not constitute a cross-paper leaderboard or assert that unlisted capabilities are absent. Prior work already establishes graph memory for EQA, temporal memory across episodes, and learned textual state updates.

Study	Memory and evaluated task	Relation to this study
GraphEQA (Saxena et al., 2025)	Hierarchical 3D scene graphs and image memory for active EQA in indoor environments.	Our main comparison fixes recorded observations and trains answerers under a blank-image condition.
Mind Palace (Ginting et al., 2025)	Temporally indexed world instances, viewpoints, and working memory for long-term active EQA.	Our memory is within an episode; cross-episode recall and exploration trade-offs are not evaluated.
EPM (Parashar et al., 2026)	Learned updates to textual object-centric state for dynamic perception and mobile-manipulation planning.	We study supervised question answering and alternative summaries/readout interfaces rather than learning memory-update operations.

GraphEQA (Saxena et al., 2025) and Mind Palace (Ginting et al., 2025) make structured and temporal embodied memory direct prior work. Planning with an Embodied Learnable Memory (Parashar et al., 2026) establishes a related role for learned textual state. The present evidence concerns a narrower interface comparison. Differences in tasks, evidence budgets, perception, and training prevent importing their published accuracies into our ablation tables.

Q SCOPE AND TRANSFER

Varying the record tests sensitivity to particular components. Re-rendering the same memory at nine budgets gives no contrast whose interval excludes zero, and removing *every* spatial relation costs 0.75 points with an interval spanning zero. Room-grouped object lists and frame stamps remain candidate sources of the benefit; the sweep does not isolate their contributions. This is consistent with the route-history pattern, but does not identify a common reasoning operation across those families. Both routes are built from the same graph; only one of them presents the temporal attributes in a form the pretrained model already reads.

What the record can distinguish. The concentration of Global-count errors points to a distinction between simultaneous counts and instance identity. With ideal instance-identified observations $O_t(r, c)$,

$$\max_t |O_t(r, c)| \;\le\; \left| \bigcup_t O_t(r, c) \right|. \tag{25}$$

The left side summarizes the largest visible count, as approximated by the text renderer; the right side counts distinct instances across the route. For an idealized example, suppose one chair appears in an early view and a different chair appears later, with no view containing both. The simultaneous maximum is one; the distinct-instance count is two. Summing per-frame counts can instead count repeated views of the same chair twice. This representational distinction motivates an instance-aware extension; it is not a calibrated bound on noisy predicted counts or a diagnosis of individual test errors. Likewise, first/last sightings omit intermediate visibility changes.

Reference-label scope also affects the interpretation of Global-count errors. The label audit in Appendix G reports Global-count accuracies from 17.4% to 44.4% against a 28.0% chance level when annotators see every frame. The individual intervals and the majority-vote interval at 31.2% include chance. The reference answer differs from a majority of annotators on 5 of 25 items. The reference rule is computed over the sampled evidence strip while the question asks about the whole house, and the strip is not recoverable from the released records. Some of the residual error in this family is therefore a property of the questions rather than of the memory representation, and the two causes cannot be separated with the present labels. Evaluating an instance-aware extension therefore requires a reference rule aligned with the question scope. A future matched-memory comparison should vary these attributes and use references aligned with question scope.

Requirements for transfer beyond simulation. All reported evaluations use simulation; physical-robot transfer remains to be tested. The memory pipeline processes 256×256 RGB frames with a prompted vision-language model and constructs scene-graph records without privileged simulator state. Applying it to real egocentric video requires assessing perception quality. The benchmark references, by contrast, come from the simulator's per-frame visibility log. A real-trajectory benchmark would require instance-level annotation or a separately validated perception-based reference, with uncertainty in those labels reported.

The interface ordering may depend on the environment, perception quality, and answerer. Mean pooling is one possible explanation for the graph result, but it is not isolated from the encoder, node-level attention, and optimization. Evaluating the ordering and absolute accuracies outside simulation therefore requires real trajectories with reliable reference annotations.

Scope and implications. The evidence concerns answering over recorded trajectories in one simulated benchmark, with two seeds in every configuration on both backbones. The main comparisons hold answer images blank and share source observations, while comparing complete configurations with their filters, prompts, and training schedules. Paired intervals and family analyses are exploratory. Image-free controls characterize question priors. The route-history group has lower blind accuracy, but options-only Order performance still exceeds chance, and low blind accuracy does not guarantee a memory benefit. Excluding Presence-change preserves the text-over-no-memory gain, and excluding both audited families (222 of the 800 questions) leaves gains of 5.80 points on MomaGraph-R1 and 3.46 on Gemma-3-4B over the remaining 578, both intervals excluding zero. Autonomous evidence acquisition remains a separate task (Ren et al., 2024; Jiang et al., 2025) (Table 24).

Q.1 Extended concluding discussion

An agent that has walked through a house holds information its current view no longer contains, and whether a vision-language model can use that information depends on how it is handed over. We study that question directly: one episodic record, built once from predicted scene graphs, presented to the answerer through two different interfaces. Room to Remember compares how pretrained vision-language models read episodic scene-graph records. Trained graph-derived text improves accuracy on both evaluated backbones (by 6.62 points on MomaGraph-R1 across two seeds, and by a smaller 3.00 points on Gemma-3-4B) and scores highest among the tested interfaces, with gains concentrated in route-history families. Learned graph access changes which questions are answered correctly but adds no observed aggregate advantage, across four matched contrasts spanning two backbones and two seeds each. Two things make that pair of results more than a ranking of two implementations. The gain is not uniform: it is worth 18.9 points on route-history questions with little measured change on the rest, so any account of it has to explain the null as well as the effect. One graph configuration failed to converge; changing its gate schedule repaired the collapse, and the recovered run still did not beat its control at either seed. This addresses the observed failure without ruling out undertraining or alternative graph readers. The study supports a concrete baseline for embodied memory: train the answerer to read the structured record, then evaluate richer interfaces through matched answer changes and the information each representation preserves. We release the benchmark, the episode records, and the per-question predictions behind every contrast, including the families whose reference labels our own audit finds unreliable.