LEARNING WHEN TO RECALL AND WHERE TO SEARCH FOR LIFELONG MULTI-MODAL NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Lifelong multi-modal navigation requires reusing past observations under changing goals. Yet storing observations selectively or in compressed form can omit details that later requests need, and retrieved evidence must still be interpreted to decide whether to revisit a known location or explore further. We introduce RecallNav, which separates goal-agnostic observation storage from goal-conditioned memory access and learns when to recall and where to search. It retains original observations across goals and uses a frozen multimodal encoder to retrieve a small set of candidate frames. A multimodal large language model (MLLM) reads retrieved, current, and frontier views within a bounded policy context and learns from expert decisions to ground targets or select destinations for further search. Explicit recall supervision ties return decisions to supporting retrieved views. Even with a 3B MLLM backbone, RecallNav outperforms all compared baselines in navigation success on every GOAT-Bench validation split. It also surpasses the reported GPT-4o-based snapshot baselines in both success rate and path efficiency on Val-Unseen. On LMEE-Bench, the same 3B policy exceeds the 7B memory-based systems in both navigation success and path efficiency. The same RGB archives support later question answering through a separate pretrained MLLM without QA-specific fine-tuning. Ablations show that goal-relevant memory access and explicit recall supervision each contribute to navigation performance.

1 INTRODUCTION

Embodied agents operating in homes and workplaces must respond to successive requests while building familiarity with their surroundings. An observation collected during one task can help locate an object requested much later, reducing the need to search the same environment again. Lifelong multi-modal navigation captures this capability through sequences of goals specified by categories, natural-language descriptions, and instance images (Khanna et al., 2024). As vision-language models improve at interpreting visual context, how past experience is preserved and accessed becomes a central question for lifelong navigation. For example, recent approaches use object-centric memories (Chang et al., 2024), visual snapshots and scene graphs (Yang et al., 2025; Wang et al., 2026a), or compressed visual histories (Hu et al., 2026) to make earlier experience available to a navigation policy.

However, making earlier experience available does not ensure that it can be reused reliably under a new goal. Selection and compression make history tractable, but they can leave the policy without the observations or visual details it needs to distinguish a later target (Chang et al., 2024; Yang et al., 2025; Hu et al., 2026). A lamp passed while searching for a sink may later become an image goal, identified by an appearance detail that the policy context no longer holds. Learned retrieval can improve access to relevant memories (Wang et al., 2026b), but it can only surface evidence that the memory it searches contains. Even when a relevant view is retrieved, the policy must decide whether it shows the requested instance and whether a return visit is warranted. Training-free snapshot systems rely on prompting for such decisions (Yang et al., 2025; Wang et al., 2026a; Chen et al., 2026), while Modular GOAT uses fixed matching criteria (Chang et al., 2024). Overlooking a valid earlier sighting can lead to repeated search, and accepting a look-alike can trigger an unnecessary

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

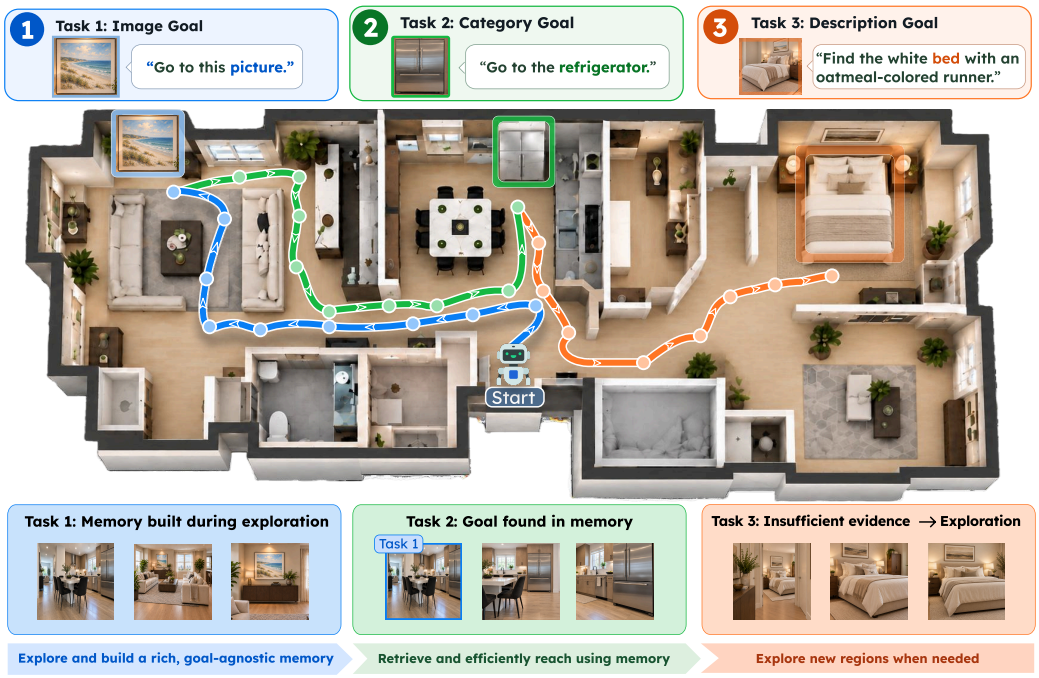

Figure 1: **Learning when to recall and where to search across changing goals.** While pursuing an image goal (Task 1), the agent retains observations for future requests. When asked to find a refrigerator (Task 2), it reads an earlier observation and returns to the target. For the subsequent description goal (Task 3), the available evidence does not support grounding the requested bed, so it selects a destination for further search.

return. The challenge is therefore to preserve experience so it can be reinterpreted under changing goals, and to learn when the available evidence supports recall rather than further search.

We introduce **RecallNav**, which separates the experience retained across goals from the evidence read for each request, and learns how that evidence should guide navigation. It stores original observations with their depth and camera pose, and it does not decide in advance which details later goals will require. A frozen multimodal encoder indexes each frame once and retrieves a small set of candidate frames for the current goal. The descriptors are only an index: the policy reads the retrieved RGB images themselves, and unselected frames stay in the archive for later requests.

To turn retrieved evidence into navigation decisions, a multimodal large language model (MLLM) reads the retrieved frames alongside the goal, the current panorama, and frontier views with spatial metadata, all within a bounded policy context. It learns from expert decisions to ground the target in a current or retrieved frame, or to select a frontier for further search. Each recall example ties the return decision to a supporting target view among the retrieved frames, so the policy learns to act on visual evidence rather than on retrieval similarity alone. Figure 1 illustrates these decisions: an observation collected during the image-goal search supports returning to the refrigerator, whereas the available evidence does not support grounding the requested bed, so the policy explores further.

Across three open 3B–4B MLLM backbones, RecallNav achieves higher navigation success than all compared baselines on GOAT-Bench (Khanna et al., 2024) and LMEE-Bench (Wang et al., 2026b). With the same Qwen2.5-VL-3B backbone as AstraNav-Memory (Hu et al., 2026), it improves success on every GOAT-Bench split and surpasses the reported GPT-4o snapshot baselines on Val-Unseen. On LMEE-Bench, it outperforms the re-evaluated 7B baselines in both navigation success and path efficiency. The same RGB archives also support later question answering through a separate pretrained MLLM without QA-specific fine-tuning. Controlled ablations show that goal-relevant memory access and explicit recall supervision each contribute to navigation performance. RecallNav thus improves navigation by learning to use retrieved experience, and its archives preserve visual evidence for later questions.

Our contributions are as follows:

- We introduce RecallNav, a lifelong navigation framework that separates goal-agnostic observation storage from goal-conditioned memory access. It preserves original observations across goals and provides retrieved frames for reinterpretation within a bounded policy context.
- We train an MLLM policy on expert decisions with explicit recall supervision, so that it reads retrieved, current, and frontier views and decides when to recall a target and where to search further.
- Across three 3B–4B MLLM backbones, RecallNav outperforms all compared baselines in navigation success on GOAT-Bench and LMEE-Bench, and the same RGB archives support subsequent question answering through a separate pretrained MLLM without additional fine-tuning.

2 Related Work

2.1 Lifelong Embodied Navigation

Navigation goals can be specified by an object category (Batra et al., 2020), an image of a place or an instance (Zhu et al., 2017; Krantz et al., 2022), or a natural-language description (Qi et al., 2020; Sun et al., 2024). Vision-and-language navigation instead provides instructions describing a route (Anderson et al., 2018). Lifelong navigation asks how experience collected for earlier requests can support subsequent ones. MultiON (Wani et al., 2020) evaluates sequential object goals, while Iterative VLN (Krantz et al., 2023) and GSA-VLN (Hong et al., 2025) study experience reuse and adaptation across repeated navigation in a scene. GOAT-Bench (Khanna et al., 2024) combines category, language, and image goals within an episode; LMEE-Bench (Wang et al., 2026b) also evaluates questions about the explored environment. Within this setting, RecallNav studies how observations collected for past goals can inform target grounding and further search under a new request.

2.2 Visual Memory for Experience Reuse

Visual memories differ in how they represent past observations and expose them to a navigation policy. Feature maps, instance memories, and scene graphs aggregate observations into spatially indexed features or entities (Huang et al., 2023; Chang et al., 2024; Busch et al., 2025; Gu et al., 2024). Snapshot-based and image-supported graph methods provide selected views and spatial context for scene reasoning (Yang et al., 2025; Wang et al., 2026a; Huang et al., 2026). Streaming and implicit-memory policies control the cost of processing history through token pruning, bounded caches, or learned visual compression (Wei et al., 2026; Zeng et al., 2026; Hu et al., 2026). External retrieval offers query-dependent access to stored descriptions or image–object memories (Anwar et al., 2025; Wang et al., 2026b). Although these methods support experience reuse, their memory representations and access mechanisms can limit the visual details available for new goals. RecallNav separates goal-agnostic observation storage from goal-conditioned memory access: it preserves RGB observations and retrieves original frames for reinterpretation within a bounded policy context.

2.3 Foundation Models for Navigation Decisions

Training-free approaches use pretrained models without navigation-specific policy training, prompting LLMs with detected object or room labels (Zhou et al., 2023; Shah et al., 2023) or textual descriptions of observations and navigation history (Zhou et al., 2024b) to guide subsequent actions. Multimodal prompting extends this interface to images, combining visual observations with topological maps, memory snapshots, or frontier views (Chen et al., 2024; Yang et al., 2025; Wang et al., 2026a; Huang et al., 2026). Training-based approaches instead learn navigation through instruction–observation–action supervision (Zheng et al., 2024; Zhou et al., 2024a), with video-based policies scaling this learning to large collections of egocentric navigation trajectories (Zhang et al., 2024; 2025; Cheng et al., 2025). Memory-based policies further learn grounding and frontier selection over spatial object queries (Zhu et al., 2025), navigation from compressed visual histories (Hu et al., 2026), or active memory retrieval and exploration through reinforcement learning (Wang et al., 2026b). Within this line of work, RecallNav's MLLM reads goal-relevant original frames from a

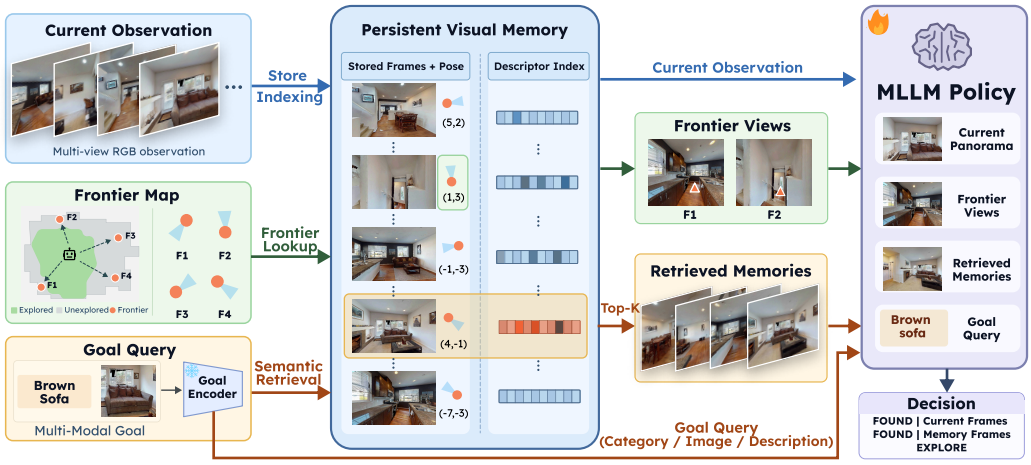

Figure 2: **RecallNav overview.** Goal-agnostic storage preserves observations across successive goals. A frozen multimodal encoder retrieves original frames for the current request, and a learned MLLM reads them with current and frontier views to choose between target grounding and further search. Captured depth and pose convert grounding into a reachable destination.

goal-agnostic archive and learns to choose between grounding a target and selecting a destination for further search.

3 Method

RecallNav learns to turn visual experience collected for past goals into informed navigation decisions for a new request. It retains observations independently of the goal, retrieves a small set of original frames relevant to the current request, and trains an MLLM to choose between grounding a target and selecting a destination for further search (Figure 2).

3.1 Problem Formulation

In lifelong multi-modal navigation (Khanna et al., 2024), an agent searches for a sequence of goals in an indoor scene. We call the search for one goal a *subtask* and the sequence an *episode*. The goal g^k of subtask k is an object category, a natural-language description, or an image of a target instance. Future goals are unknown until their subtasks begin, and each subtask starts where the preceding one ends. The objective is to reach the requested targets efficiently by using both current observations and experience from earlier searches.

In our setting, the simulator provides RGB-D observations and camera poses. Following MTU3D (Zhu et al., 2025) and AstraNav-Memory (Hu et al., 2026), we also use simulator-provided navigability for frontier construction and low-level movement. Visual memory and exploration state persist across subtasks. This setting distinguishes *goal-agnostic observation storage*, which preserves observations before their future relevance is known, from *goal-conditioned memory access*, which exposes relevant past observations to the policy under a new request.

3.2 Goal-Agnostic Observation Storage and Goal-Conditioned Memory Access

Observation archive. At decision t, the archive $\mathcal{A}_t$ contains observations acquired during movement and at decision locations. Each archived observation stores its RGB image X_i, corresponding depth D_i, camera pose T_i, and capture time τ_i. The observation storage rule does not depend on the current goal, so observations encountered while pursuing one target remain available for later requests. The archive grows throughout the episode, and retrieval bounds how much of it the policy reads at each decision.

Goal-conditioned access. A frozen multimodal encoder maps archived RGB images and the current goal to descriptors z_i and q^k in a shared embedding space. We rank historical frames by cosine similarity, $s(g^k, X_i) = (q^k)^\top z_i / (\|q^k\|_2 \|z_i\|_2)$, and retrieve up to $K = 4$ frames from the current episode, excluding the current panorama. The selected frames are ordered by capture time to form the memory context $\mathcal{M}_t^k$. The same frozen encoder and retrieval rule construct the training contexts and select memories at inference. Appendix B specifies the archive, query construction, and encoder settings.

Reading original observations. The retrieval descriptors only select observations and are not passed to the policy; the navigation MLLM reads the retrieved RGB frames themselves, with their capture metadata. It can therefore reinterpret their appearance and context under the current request. Retrieval supplies candidate observations, and the learned policy decides whether the available evidence supports target grounding or calls for further search.

3.3 Evidence-Grounded Navigation Decisions

Decision context. The policy context C_t comprises the goal g^k, retrieved memories $\mathcal{M}_t^k$, the current panorama $\mathcal{P}_t$, and frontier candidates $\mathcal{F}_t$, together with the current pose and associated spatial metadata. The panorama contains four views at 90° intervals. Frontiers separate observed from unobserved navigable space and are associated with captured camera views that provide context for further search. The MLLM receives RGB images but not depth, which is used only for geometric localization. Appendices B.1 and E detail the prompt and frontier construction.

Grounding or further search. The MLLM jointly selects an action type and its arguments:

$$a_t = \pi_\theta(C_t), \qquad a_t \in \{\texttt{EXPLORE}(j),\ \texttt{FOUND}(i, b)\}. \tag{1}$$

Here, j identifies an available frontier, while i identifies a displayed memory or current-panorama image and b is the target's 2D bounding box in that image. An exploration action selects where to continue searching; a grounding action ties the target prediction to a specific observation with a known capture pose. The policy thus learns both which decision to make and the visual grounding or destination selection needed to execute it.

Action execution and memory recall. For a grounding action, the depth captured with the evidence image and its pose yield a 3D target estimate, from which a geometric controller selects a nearby reachable destination. An exploration action instead uses the chosen frontier as its destination. This separates image-space grounding from depth-based destination computation, so the policy relies less on the backbone's learned coordinate prediction. Grounding in memory triggers a return visit followed by a new panorama and decision; we call this a *recall*. The agent stops only after reaching a destination constructed from current-view grounding, so every recall is followed by a fresh panorama and a new decision. Exploration likewise leads to a new decision after movement. Appendix C gives movement and fallback details.

3.4 Learning from Retrieved Experience

Expert decisions. We construct decision-level supervision from expert rollouts on GOAT-Bench training scenes in HM3D (Ramakrishnan et al., 2021; Yadav et al., 2023). Using instance semantics, valid goal viewpoints, and navigability, the expert prioritizes a supported current-view detection, then a useful memory recall, and otherwise exploration toward the goal. Instance semantics and goal-viewpoint annotations are used only for supervision; at inference, destinations come from the policy's own grounding predictions. Appendix A describes rollout collection and dataset construction.

Retrieval-consistent supervision. For each recorded decision, we retrieve memories only from observations available up to that point and express the expert action in terms of the context actually presented to the navigation policy. If an expert memory recall has no supporting target view among the retrieved frames, we substitute the expert's alternative exploration action when one is available; otherwise, the example is omitted. The supervised action then matches the evidence the policy can see, and the policy is never asked to recall a target from an observation it did not receive.

Policy optimization. Let $\mathcal{D}$ contain the resulting context–action pairs (C, y), where y is the serialized action output. We apply supervised fine-tuning (SFT) to the MLLM using next-token cross-entropy on the action tokens:

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\mathbb{E}_{(C,y)\sim\mathcal{D}}\left[\sum_{\ell=1}^{|y|} \log p_\theta(y_\ell \mid C, y_{<\ell})\right], \tag{2}$$

where p_θ is the MLLM's token distribution. We update the language model and vision-language aligner while freezing the vision encoder; the retrieval encoder remains fixed. All evaluated backbones share the training examples and retrieval rule, with model-specific bounding-box conventions. Appendix A.1 provides checkpoints and optimization settings.

4 Experiments

We evaluate whether visual experience collected across goals can support effective navigation decisions and remain useful for later requests. Benchmark comparisons assess system-level performance, while controlled ablations examine access to goal-relevant history, persistence across goals, and learning to act on the available evidence.

4.1 Experimental Setup

Benchmarks and metrics. We evaluate all episodes in the Val-Seen, Val-Seen-Synonyms, and Val-Unseen splits of GOAT-Bench (Khanna et al., 2024), and navigation followed by memory-based question answering on LMEE-Bench (Wang et al., 2026b). The latter uses all 166 released test episodes, comprising 821 navigation goals and 405 questions after the description-length filter provided with the benchmark. We report navigation success rate (SR), success weighted by path length (SPL), open-ended QA quality (Score), and multiple-choice accuracy (Acc), all on a 0–100 scale. RecallNav requires a declared stop less than 1 m geodesically from a valid goal viewpoint. To mitigate the skew in correct-answer positions in the released questions, we re-evaluate multiple-choice QA using the same reordered choices with balanced answer positions for all methods (Appendix I). Open-ended answers are scored with a common evaluation model (Appendix G).

Baselines. On GOAT-Bench, the main comparisons are the training-free snapshot systems 3D-Mem, SCOPE, and HGR (Yang et al., 2025; Wang et al., 2026a; Chen et al., 2026) and the navigation-trained AstraNav-Memory (Hu et al., 2026); Table 1 also lists earlier navigation baselines and MTU3D (Zhu et al., 2025), which we re-evaluate on all episodes. On LMEE-Bench, we compare against 3D-Mem, RA-Mem (Wang et al., 2026b), and MemoryExplorer (Wang et al., 2026b), which explore and then answer questions from their collected memories. Table 2 separates their paper-reported navigation results from our re-evaluations. These comparisons are at the system level, under each method's stated sensing, control, and stopping protocols, and do not isolate the backbone (Appendix G.1).

Implementation details. We train Qwen2.5-VL-3B (Bai et al., 2025b), Qwen3-VL-4B (Bai et al., 2025a), and Qwen3.5-4B (Qwen Team, 2026) on the same GOAT-Bench expert decisions and evaluate the resulting checkpoints on both benchmarks. The frozen Qwen3-VL-Embedding-2B (Li et al., 2026) retriever selects up to $K = 4$ historical images; training settings are given in Appendix A.1. For the QA comparison in Table 2, RecallNav, 3D-Mem, and RA-Mem use pretrained Qwen2.5-VL-7B without additional fine-tuning, whereas MemoryExplorer retains its released fine-tuned model (Appendix G.1). All RecallNav variants answer from three images retrieved from their respective navigation archives.

4.2 Benchmark Results

Strong navigation across backbones. Table 1 reports results on all three GOAT-Bench splits. With the same Qwen2.5-VL-3B backbone as AstraNav-Memory, RecallNav reaches much higher success on every split with competitive path efficiency. This policy also exceeds the reported GPT-4o snapshot baselines in both SR and SPL on Val-Unseen, so the gains do not depend on a newer

Table 1: GOAT-Bench results on all episodes of each validation split. †Our re-evaluation on all episodes; other baseline values are paper-reported.

Method	Navigation MLLM	Val-Seen		Val-Seen-Syn.		Val-Unseen	
		SR	SPL	SR	SPL	SR	SPL
Modular CLIP on Wheels (Gadre et al., 2023)	–	14.8	8.7	18.5	11.5	16.1	10.4
Modular GOAT (Chang et al., 2024)	–	26.3	17.5	33.8	24.4	24.9	17.2
SenseAct-NN Monolithic (Khanna et al., 2024)	–	16.8	9.4	18.5	10.1	12.3	6.8
SenseAct-NN Skill Chain (Khanna et al., 2024)	–	29.2	12.8	38.2	15.2	29.5	11.3
MTU3D† (Zhu et al., 2025)	–	60.0	36.0	58.7	32.3	50.4	29.0
AstraNav-Memory (Hu et al., 2026)	Qwen2.5-VL-3B	65.5	49.0	66.8	54.7	62.7	56.9
3D-Mem (Yang et al., 2025)	GPT-4o	–	–	–	–	62.9	44.7
SCOPE (Wang et al., 2026a)	GPT-4o	–	–	–	–	66.8	46.5
HGR (Chen et al., 2026)	GPT-4o	–	–	–	–	64.1	50.1
RecallNav	Qwen2.5-VL-3B	71.0	53.9	73.5	54.2	73.3	56.5
RecallNav	Qwen3-VL-4B	75.5	58.6	78.8	60.1	80.7	62.9
RecallNav	Qwen3.5-4B	**76.6**	**59.6**	**81.4**	**63.3**	**81.0**	**63.2**

Table 2: **LMEE-Bench navigation and QA.** Paper-reported values are taken from LMEE-Bench (Wang et al., 2026b); the remaining columns report our evaluation. QA uses the same re-ordered answer choices with balanced correct-answer positions and the same evaluation model for open-ended answers. Listed backbones are used for navigation. QA models are specified in Appendix G.1.

Method (MLLM backbone)	Paper-reported		Our evaluation			
	SR	SPL	SR	SPL	Score	Acc
3D-Mem (Qwen2.5-VL-7B) (Yang et al., 2025)	17.7	8.1	18.8	7.4	36.8	42.2
RA-Mem (Qwen2.5-VL-7B) (Wang et al., 2026b)	20.5	11.4	20.1	8.3	42.6	51.9
MemoryExplorer (Qwen2.5-VL-7B) (Wang et al., 2026b)	22.1	14.3	23.4	14.1	43.5	53.6
RecallNav (Qwen2.5-VL-3B)	–	–	53.1	35.2	45.3	57.8
RecallNav (Qwen3-VL-4B)	–	–	57.6	40.9	**48.7**	**58.3**
RecallNav (Qwen3.5-4B)	–	–	**61.1**	**41.4**	46.7	57.0

backbone. Qwen3 and Qwen3.5 improve the results further. Replacing the main frontier map with depth-integrated TSDF frontiers lowers performance, but all three navigation policies stay above the reported GPT-4o snapshot baselines on all Val-Unseen episodes (Appendix E).

Navigation experience remains useful for later questions. Table 2 evaluates navigation and the later use of the observations collected along each agent's trajectory. With 3B–4B MLLM backbones, RecallNav outperforms the re-evaluated 7B memory-based baselines by a wide margin in both navigation success and path efficiency. The paper-reported baseline SRs are broadly consistent with our re-evaluations. RecallNav's unmodified RGB archives also support a separate QA model used without additional fine-tuning. Across MLLM backbones, they yield open-ended QA scores of 45.3–48.7 and multiple-choice accuracies of 57.0–58.3%. Navigation and QA rankings differ, though: Qwen3.5 has the highest navigation success, whereas Qwen3 gives the best QA results. Lifelong experience should therefore be judged both by the destinations it helps reach and by the evidence it provides for later requests.

From visual evidence to goal-specific destinations. Figure 3 illustrates why category relevance and prior exposure alone do not guarantee an appropriate destination. For the carpet image goal, RecallNav distinguishes the requested instance from other carpets, whereas SCOPE reaches a different instance. For the described mirror, RecallNav grounds the target, while SCOPE misses the mirror and selects a nearby curtain. For the later calendar request, RecallNav uses an observation from its earlier route to return directly to the target; SCOPE passes its vicinity without recognizing it



Figure 3: **Visual evidence for instance recognition and recall.** Image-, description-, and category-goal examples illustrate distinguishing a carpet instance, grounding a mirror, and returning to a calendar encountered earlier. AstraNav-Memory insets show the observation at its final position, not an identified frame supporting its decision. The calendar request follows an intervening carpet subtask.

Table 3: **Memory access ablations.** Qwen3.5-4B on GOAT-Bench Val-Unseen.

Variant	SR↑	SPL↑
RecallNav (full)	**81.0**	**63.2**
Recent-K	55.3	41.4
Category pre-filter	76.2	58.6
State reset	78.6	53.4

Table 4: **Decision and grounding.** Qwen3.5-4B on GOAT-Bench Val-Unseen.

Variant	SR↑	SPL↑
RecallNav (full)	**81.0**	63.2
No SFT	57.0	32.3
Recall → current	62.5	47.8
Direct coordinates	79.2	**65.4**

and eventually times out. AstraNav-Memory also ends near categorically related or visually similar distractors in these examples; its insets show only its endpoint observations. The examples show two linked requirements of experience reuse: interpreting visual details under the current request, and turning a useful earlier sighting into a destination.

4.3 Ablations and Analysis

Tables 3 and 4 use Qwen3.5 on all 360 GOAT-Bench Val-Unseen episodes and 2,669 goals. Memory-access variants keep the trained policy and controller fixed. The policy without navigation fine-tuning uses the original backbone with explicit action examples; Direct coordinates is separately trained on the same decision examples with the same optimization settings. Appendices B.3, B.1, and D detail these interventions.

Goal relevance matters beyond recency. Replacing goal-conditioned retrieval with the most recent K frames lowers SR and SPL by 25.7 and 21.8 points at the same $K = 4$ historical-image budget (Table 3). Because both variants receive the same number of memory images, the gap comes from which history the policy reads: recent observations support local continuity but can omit a relevant target seen earlier. This favors separating retained experience from the active context and

selecting past observations according to the current goal. The effect is much larger than the variation across the evaluated semantic retrieval encoders (Appendix B.2).

Intermediate semantic filtering can restrict useful evidence. Restricting retrieval to frames associated with up to ten object classes selected by a detector (Appendix B.3) and an MLLM lowers SR and SPL by 4.8 and 4.6 points, respectively (Table 3). This filtering step decides which observations the navigation MLLM can inspect before it recognizes and grounds the target. The result shows a cost of intermediate semantic filtering: shrinking the candidate set can also remove visual evidence the downstream decision needs. The intervention changes only access to stored observations; the storage rule is unchanged.

Persistent experience reduces the cost of rediscovery. Resetting earlier-subtask memory access alone would leave previously explored regions marked as explored and unavailable as destinations for renewed frontier-based search. We therefore reset exploration state alongside memory access at each subtask transition. This joint reset lowers Qwen3.5 SR by 2.4 points but SPL by 9.8 points (Table 3). The larger efficiency penalty shows that persistence matters even when many targets can eventually be found again. Unlike Recent-K, State reset still retrieves relevant observations accumulated within the current subtask, which is consistent with its stronger performance. The same pattern holds for Qwen2.5 and Qwen3, with larger losses in both metrics (Appendix B.3).

Learning to act on retrieved experience. Policy fine-tuning improves SR and SPL by 24.0 and 30.9 points over a pretrained policy prompted with explicit action examples (Table 4). Both policies use the same retrieval procedure and geometric controller, so the gain reflects navigation-specific learning on top of access to relevant observations. Appendix B.1 reports the prompts and parsing rates for both policies.

Memory-recall supervision contributes beyond current-view grounding. Replacing all 6,250 memory-recall training examples with the same number of current-view grounding examples (Appendix A) lowers SR and SPL by 18.5 and 15.4 points (Recall $\rightarrow$ current in Table 4). The replacement keeps the total number of training examples, the numbers of `EXPLORE` and `FOUND` targets, and the number of optimization updates unchanged. Inference uses the same retrieval procedure, action interface, and controller, and memory-grounded actions remain available. This result supports the role of explicit recall supervision in learning to turn retrieved evidence into navigation decisions. Together with the memory-access ablations, it highlights two complementary contributions to experience reuse: providing access to goal-relevant observations and learning how those observations should guide action.

Depth-based destination computation reduces reliance on metric prediction. Direct coordinate prediction trades slightly lower success for higher SPL with Qwen3.5, whereas it loses much more success with Qwen2.5 (Appendix D). This backbone-dependent trade-off is why we derive metric destinations from depth and pose instead of relying entirely on learned coordinate prediction. Appendix F separately analyzes the resource trade-off of retaining a growing archive while limiting the visual context read at each decision.

5 Conclusion

We presented RecallNav, a framework for lifelong multi-modal navigation that combines goal-agnostic observation storage, goal-conditioned retrieval, and an MLLM policy trained to ground navigation actions in current and historical observations. Experiments on GOAT-Bench and LMEE-Bench demonstrate substantial improvements in navigation success across three backbones. Ablations show that goal-relevant history contributes beyond recent observations, while explicitly supervising memory-recall actions provides benefits that additional current-view grounding examples do not recover. Persistent experience also reduces the cost of rediscovery, and the same navigation archives support later questions through a separate pretrained QA model. Together, these findings show that useful memory requires both access to relevant evidence and learning how that evidence should guide action. Our evaluation remains limited to finite episodes in static scenes, and the archive has no fixed storage limit. Future work should address bounded storage that preserves distinctive visual evidence and memory updates in dynamic environments.

AI USE STATEMENT

In this work, we used generative AI tools to assist with code implementation. We did not use generative AI tools for synthetic data generation, development of theoretical models or conceptual frameworks, hypothesis formulation, experimental design or research methodology, or interpretation of results, and the remaining disclosure categories are not applicable to this work. Additionally, we used generative AI tools to draft portions of the manuscript and to polish the writing. The first author verified all AI-assisted code for correctness, and we carefully reviewed and revised all AI-assisted text to ensure its accuracy, consistency, and fidelity to our original scientific claims. We take full responsibility for the final content of this work, including any text, code, or claims produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, the main paper describes RecallNav's goal-agnostic observation storage, goal-conditioned memory access, and evidence-grounded navigation policy. Appendix A details expert rollout collection, training-example construction, backbone initializations, and optimization hyperparameters. Appendices B, C, and E specify memory retrieval, prompt and output formats, image preprocessing, target localization, action execution, and frontier mapping. Appendix G documents benchmark splits, sensing and control settings, navigation and QA evaluation protocols, and baseline re-evaluation, with answer-choice balancing detailed in Appendix I. We also provide ablations and sensitivity analyses, as well as inference-time and storage measurements in Appendix F. Our source code, configuration files, and supporting materials will be publicly released upon publication.

REFERENCES

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3674–3683, 2018.

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 *IEEE International Conference on Robotics and Automation*, pp. 2838–2845, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025a. URL https://arxiv.org/abs/2511.21631.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025b. URL https://arxiv.org/abs/2502.13923.

Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. Objectnav revisited: On evaluation of embodied agents navigating to objects. arXiv preprint arXiv:2006.13171, 2020. URL https://arxiv.org/abs/2006.13171.

Finn Lukas Busch, Timon Homberger, Jesús Ortega-Peimbert, Quantao Yang, and Olov Andersson. One map to find them all: Real-time open-vocabulary mapping for zero-shot multi-object

navigation. In *IEEE International Conference on Robotics and Automation*, pp. 14835–14842, 2025.

Matthew Chang, Theophile Gervet, Mukul Khanna, Sriram Yenamandra, Dhruv Shah, So Yeon Min, Kavit Shah, Chris Paxton, Saurabh Gupta, Dhruv Batra, Roozbeh Mottaghi, Jitendra Malik, and Devendra Singh Chaplot. Goat: Go to any thing. In *Robotics: Science and Systems*, 2024.

Jiaqi Chen, Bingqian Lin, Ran Xu, Zhenhua Chai, Xiaodan Liang, and Kwan-Yee Wong. Mapgpt: Map-guided prompting with adaptive path planning for vision-and-language navigation. In *Annual Meeting of the Association for Computational Linguistics*, pp. 9796–9810, 2024.

Peixin Chen, Guoxi Zhang, Jianwei Ma, and Qing Li. Hypothesis graph refinement: Hypothesis-driven exploration with cascade error correction for embodied navigation. In *European Conference on Computer Vision*, 2026.

An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Zaitian Gongye, Xueyan Zou, Jan Kautz, Erdem Biyik, Hongxu Yin, Sifei Liu, and Xiaolong Wang. Navila: Legged robot vision-language-action model for navigation. In *Robotics: Science and Systems*, 2025.

Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-World: Real-time open-vocabulary object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16901–16911, 2024.

Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. CoWs on pasture: Baselines and benchmarks for language-driven zero-shot object navigation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23171–23181, 2023.

Qiao Gu, Ali 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 *IEEE International Conference on Robotics and Automation*, pp. 5021–5028, 2024.

Haodong Hong, Yanyuan Qiao, Sen Wang, Jiajun Liu, and Qi Wu. General scene adaptation for vision-and-language navigation. In *International Conference on Learning Representations*, pp. 69956–69983, 2025.

Junjun Hu, Xinda Xue, Botao Ren, Minghua Luo, Jintao Chen, Haochen Bai, Liangliang You, and Mu Xu. Astranav-memory: Contexts compression for long memory. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8097–8109, 2026.

Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *IEEE International Conference on Robotics and Automation*, pp. 10608–10615, 2023.

Xun Huang, Shijia Zhao, Yunxiang Wang, Xin Lu, Wanfa Zhang, Rongsheng Qu, Weixin Li, Yunhong Wang, and Chenglu Wen. MSGNav: Unleashing the power of multi-modal 3d scene graph for zero-shot embodied navigation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 37154–37163, 2026.

Mukul Khanna, Ram Ramrakhya, Gunjan Chhablani, Sriram Yenamandra, Theophile Gervet, Matthew Chang, Zsolt Kira, Devendra Singh Chaplot, Dhruv Batra, and Roozbeh Mottaghi. GOAT-Bench: A benchmark for multi-modal lifelong navigation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16373–16383, 2024.

Jacob Krantz, Stefan Lee, Jitendra Malik, Dhruv Batra, and Devendra Singh Chaplot. Instance-specific image goal navigation: Training embodied agents to find object instances. arXiv preprint arXiv:2211.15876, 2022. URL `https://arxiv.org/abs/2211.15876`.

Jacob Krantz, Shurjo Banerjee, Wang Zhu, Jason Corso, Peter Anderson, Stefan Lee, and Jesse Thomason. Iterative vision-and-language navigation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14921–14930, 2023.

Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Qwen3-vl-embedding and qwen3-vl-reranker: A unified framework for state-of-the-art multimodal retrieval and ranking, 2026. URL https://arxiv.org/abs/2601.04720.

Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A co-habitat for humans, avatars, and robots. In *International Conference on Learning Representations*, pp. 15306–15336, 2024.

Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. Reverie: Remote embodied visual referring expression in real indoor environments. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9982–9991, 2020.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763, 2021.

Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3D): 1000 large-scale 3d environments for embodied AI. In *Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *IEEE/CVF International Conference on Computer Vision*, 2019.

Dhruv Shah, Michael Robert Equi, Błażej Osiński, Fei Xia, Brian Ichter, and Sergey Levine. Navigation with large language models: Semantic guesswork as a heuristic for planning. In *Conference on Robot Learning*, pp. 2683–2699, 2023.

Xinyu Sun, Lizhao Liu, Hongyan Zhi, Ronghe Qiu, and Junwei Liang. Prioritized semantic learning for zero-shot instance navigation. In *European Conference on Computer Vision*, pp. 161–178, 2024.

Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems*, 2021.

Ningnan Wang, Weihuang Chen, Liming Chen, Haoxuan Ji, Zhongyu Guo, Xuchong Zhang, and Hongbin Sun. Expand your SCOPE: Semantic cognition over potential-based exploration for embodied visual navigation. In *AAAI Conference on Artificial Intelligence*, pp. 18620–18628, 2026a.

Sen Wang, Bangwei Liu, Zhenkun Gao, Lizhuang Ma, Xuhong Wang, Yuan Xie, and Xin Tan. Explore with long-term memory: A benchmark and multimodal llm-based reinforcement learning framework for embodied exploration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 37098–37108, 2026b.

Saim Wani, Shivansh Patel, Unnat Jain, Angel Chang, and Manolis Savva. MultiON: Benchmarking semantic map memory using multi-object navigation. In *Advances in Neural Information Processing Systems*, pp. 9700–9712, 2020.

Meng Wei, Chenyang Wan, Xiqian Yu, Tai Wang, Yuqiang Yang, Xiaohan Mao, Chenming Zhu, Wenzhe Cai, Hanqing Wang, Yilun Chen, Xihui Liu, and Jiangmiao Pang. Streamvln: Streaming vision-and-language navigation via slowfast context modeling. In *IEEE International Conference on Robotics and Automation*, 2026.

Karmesh Yadav, Ram Ramrakhya, Santhosh Kumar Ramakrishnan, Theo Gervet, John Turner, Aaron Gokaslan, Noah Maestre, Angel Xuan Chang, Dhruv Batra, Manolis Savva, Alexander William Clegg, and Devendra Singh Chaplot. Habitat-matterport 3d semantics dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4927–4936, 2023.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17294–17303, 2025.

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*, pp. 33001–33026, 2026.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision*, pp. 11975–11986, 2023.

Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. Navid: Video-based VLM plans the next step for vision-and-language navigation. In *Robotics: Science and Systems*, 2024.

Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks. In *Robotics: Science and Systems*, 2025.

Duo Zheng, Shijia Huang, Lin Zhao, Yiwu Zhong, and Liwei Wang. Towards learning a generalist model for embodied navigation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13624–13634, 2024.

Gengze Zhou, Yicong Hong, Zun Wang, Xin Eric Wang, and Qi Wu. Navgpt-2: Unleashing navigational reasoning capability for large vision-language models. In *European Conference on Computer Vision*, pp. 260–278, 2024a.

Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In *AAAI Conference on Artificial Intelligence*, pp. 7641–7649, 2024b.

Kaiwen Zhou, Kaizhi Zheng, Connor Pryor, Yilin Shen, Hongxia Jin, Lise Getoor, and Xin Eric Wang. Esc: Exploration with soft commonsense constraints for zero-shot object navigation. In *International Conference on Machine Learning*, pp. 42829–42842, 2023.

Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *IEEE International Conference on Robotics and Automation*, pp. 3357–3364, 2017.

Ziyu Zhu, Xilin Wang, Yixuan Li, Zhuofan Zhang, Xiaojian Ma, Yixin Chen, Baoxiong Jia, Wei Liang, Qian Yu, Zhidong Deng, Siyuan Huang, and Qing Li. Move to understand a 3d scene: Bridging visual grounding and exploration for efficient and versatile embodied navigation. In *IEEE/CVF International Conference on Computer Vision*, pp. 8120–8132, 2025.

Appendix

TABLE OF CONTENTS

A **Expert Data and Training** . . . 15
A.1 Optimization . . . 16
B **Visual Memory and Decision Interface** . . . 16
B.1 Input Prompt and Output Format . . . 16
B.2 Effect of the Retrieval Encoder . . . 17
B.3 Memory Access Ablations . . . 17
B.4 Effect of the Number of Retrieved Memory Images . . . 18
C **Target Localization and Action Execution** . . . 18
D **Visual Grounding versus Coordinate Prediction** . . . 19
E **Frontier Mapping and Baseline Comparisons** . . . 19
F **Inference Time and Storage Requirements** . . . 20
G **Evaluation Protocol** . . . 20
G.1 LMEE-Bench Baseline Re-evaluation . . . 21
H **GOAT-Bench Results by Goal Type** . . . 22
I **Sensitivity to Answer-Choice Order in LMEE-Bench** . . . 22
J **LMEE-Bench Results by Difficulty** . . . 23

A EXPERT DATA AND TRAINING

Expert rollouts. We collect 6,000 episodes from 136 HM3D training scenes, comprising 45,071 category, description, and image-goal subtasks. The expert prioritizes grounding in the current view when the target is visible, then memory recall, and otherwise exploration. For each acceptable instance, it fixes the reachable goal viewpoint nearest to the subtask's initial position in geodesic distance. Instance-level semantic annotations and goal viewpoints provide expert supervision; they are not used as policy inputs or destinations at inference.

Target visibility. Category goals accept any listed instance of the requested category; description and image goals require the designated instance. During rollouts, instance i is visible in frame f when the number of pixels $n_i(f)$ in its semantic mask satisfies

$$n_i(f) \geq \rho_i, \qquad \rho_i = \max\left(50, \operatorname{round}\left(0.01 \max_{v \in V_i} n_i(v)\right)\right), \tag{3}$$

where V_i contains the supplied goal viewpoints; the threshold is 50 pixels when no reference is available. Historical frames must be captured before the current panorama. Grounding requires the selected instance's reference viewpoint to lie in the observed map; memory recall also requires that no current view satisfy the visibility criterion.

Exploration and recall. The expert selects the available frontier whose representative navigable position has the smallest finite geodesic distance to a fixed target viewpoint, provided this distance is smaller than for previously visited frontier positions. When such a frontier exists, recall is preferred only if its route is more than 1 m shorter than the route through that frontier to a target viewpoint. If no such frontier exists, this margin is waived.

Training-example construction. For each training context, the retriever used at evaluation selects historical frames captured before the current panorama. Grounding labels remain tied to the expert-selected instance. A memory frame supports the label if it has a valid bounding box and a nonempty mask for the selected instance, and either was recorded as a supporting frame during the expert rollout or meets ρ_i. The bounding box is obtained from the supporting frame with the largest instance mask, with ties resolved chronologically. Labels for current-view grounding use the recorded view when its box is valid, otherwise using the panorama's largest valid instance mask without reapplying ρ_i. Recall labels without supporting retrieved frames are replaced by the recorded exploration alternative only when its finite distance uniquely matches a frontier in the final prompt within 10^{-3} m; otherwise these examples are omitted. Balancing across scenes and categories and downsampling straightforward detections after recall yield the dataset in Table 5. All navigation policies use the same decision-level examples and action labels, with model-specific box coordinates.

Table 5: Composition of the decision-level dataset at $K = 4$. Relabeled recalls are included in exploration; omitted examples are excluded from the total.

Quantity	Training
Exploration	27,153
Current-view grounding	19,558
Memory recall	6,250
Total examples	**52,961**
Unsupported recall → exploration	1,294
Unsupported recall omitted	62

Recall-to-current ablation. For the Recall → current variant in Table 4, we replace the 6,250 memory-recall examples with an equal number of examples sampled uniformly with replacement (seed 42) from the 19,558 current-view grounding examples that remain in the training set. All replacements are copies of retained training examples; no examples from separate rollouts or new unique examples are added. We retain all exploration examples, including the 1,294 relabeled recalls, preserving the total of 52,961 training examples and the `EXPLORE`/`FOUND` counts.

A.1 Optimization

We initialize from Qwen2.5-VL-3B-Instruct, Qwen3-VL-4B-Instruct, and Qwen3.5-4B. Each navigation policy is trained for one epoch with the common settings in Table 6. Equation 2 supervises assistant action tokens; prompt tokens and empty reasoning delimiters are masked, and no reasoning traces are supplied. The vision encoder and retrieval encoder remain frozen. We evaluate the final checkpoint at step 1,655 without selecting checkpoints based on benchmark performance. Image preprocessing and bounding-box coordinate formats are specified in Appendix B.1.

Table 6: Common SFT settings for the three RecallNav policies.

Parameter	Value
Trainable modules	Language model and vision–language aligner
Training examples / epochs / steps	52,961 / 1 / 1,655
Optimizer	AdamW, $\beta = (0.9, 0.95)$, $\epsilon = 10^{-8}$
Peak learning rate / schedule	2×10^{-5} / cosine
Warmup / weight decay / gradient clipping	5% / 0.1 / 1.0
GPUs / per-device batch / accumulation	4 / 1 / 8 (effective batch 32)
Precision / attention	bfloat16 / FlashAttention-2
Gradient checkpointing / sequence packing	Enabled / disabled
Model seed / data seed	1 / 42

B Visual Memory and Decision Interface

Observation archive. The archive stores observations acquired during movement and four panorama views at 90° intervals. Each observation includes RGB, depth recorded with the image, camera pose, capture time, and source metadata. The storage rule does not filter observations by the current goal, although the trajectory may depend on it. The archive and descriptor index persist across subtasks without a storage limit. Depth is used for geometric localization and is not provided as a visual input to the MLLM.

Descriptors and retrieval. The frozen Qwen3-VL-Embedding-2B (Li et al., 2026) encoder uses the instruction `Represent the user's input.` Category and description queries use their text; image queries use the goal image. Last-token pooling and ℓ_2 normalization produce 2,048-dimensional descriptors. Each archived frame is encoded once, and each goal at subtask start. Cosine similarity selects up to $K = 4$ historical frames, excluding the current panorama and without category pre-filtering. Selected frames are ordered chronologically as $M_1, \ldots, M_m$, $m \leq K$. The policy receives their original RGB images through the model's image processor; the retrieval descriptors are not passed to it. Training and inference use the same retrieval rule. Appendix B.4 evaluates the number of retrieved memory images.

B.1 Input Prompt and Output Format

Input. A single user message uses the model's native chat template, with images ordered as retrieved memories, four panorama views, an optional image goal, and up to eight frontier views: at most 17 images. Memory metadata gives frame ID, agent-relative planar pose and yaw, source, previous action, number of frames since capture, and subtask ID; empty history is marked `[NO_HISTORY]`. The prompt also includes the current pose, goal, and a frontier table with IDs, relative positions, bearings, distances, and associated views. Image goals include no category text.

Output. The SFT instruction is `Emit exactly one <NAV> action.` An `EXPLORE` action selects a frontier ID; `FOUND` selects a displayed current or memory image and a box, not a frontier image. Qwen2.5 expresses bounding boxes in pixel coordinates of the processed 728×644 image; Qwen3 and Qwen3.5 use integers normalized to $[0, 1000]$ and process images at 704×640. The controller maps boxes back to the archived 720×640 image. Example Qwen3.5 outputs from a completed rollout are:

```
<NAV><mode>EXPLORE</mode><candidate>FRONTIER_1</candidate></NAV>
```

```
<NAV><mode>FOUND</mode><evidence>current:P1</evidence>
<box><|box_start|>(493,342),(607,434)<|box_end|></box></NAV>
<NAV><mode>FOUND</mode><evidence>memory:M4</evidence>
<box><|box_start|>(776,0),(999,309)<|box_end|></box></NAV>
```

Comparison without policy fine-tuning. For Qwen3.5 without navigation fine-tuning, the final SFT instruction is replaced with examples of all three output cases, restrictions on valid image indices, the bounding-box coordinate format, and an instruction to provide no explanation. Input construction, controller, decoding, and parser are otherwise shared. Both policies use greedy decoding with thinking mode disabled, a 64-token output limit, and repetition penalty 1.0. The parser accepts native box tokens and equivalent flat or singly nested integer lists; a missing final `</NAV>` is restored only after complete action fields, without generating action content. Table 7 reports navigation performance and the fraction of outputs successfully parsed.

Table 7: Effect of policy fine-tuning for Qwen3.5-4B on all 360 GOAT-Bench Val-Unseen episodes (2,669 goals). Parsing success measures the fraction of responses accepted by the parser, not action correctness; navigation metrics are computed per subtask.

MLLM backbone	Policy	SR	SPL	Parsing success
Qwen3.5-4B	No SFT, action examples	57.0	32.3	98.6
Qwen3.5-4B	SFT, training prompt	**81.0**	**63.2**	**99.4**

B.2 Effect of the Retrieval Encoder

We replace the default retriever with SigLIP (Zhai et al., 2023) or CLIP ViT-B/32 (Radford et al., 2021), keeping the trained Qwen3.5 policy, observation storage rule, $K = 4$ memory-image limit, and controller unchanged. Table 8 reports the resulting closed-loop navigation performance.

Table 8: Effect of the retrieval encoder on all GOAT-Bench Val-Unseen episodes. Only the retrieval encoder configuration changes; the navigation policy is not retrained.

Retrieval encoder	SR↑	SPL↑
Qwen3-VL-Embedding-2B	**81.0**	**63.2**
SigLIP	78.9	61.7
CLIP ViT-B/32	77.2	59.7

B.3 Memory Access Ablations

The variants in Table 3 retain the trained policy and controller without retraining. These are closed-loop evaluations: changes to memory access can affect subsequent actions and observations, so the input contexts are not held fixed.

Recent-K. We replace goal-conditioned ranking with the four most recent historical frames, excluding the current panorama, while retaining their original images and metadata.

Category pre-filter. Inspired by the category-based pre-filtering in 3D-Mem (Yang et al., 2025), we restrict the candidate frames using detected object categories before applying the unchanged retriever. YOLO-World (Cheng et al., 2024) uses a confidence threshold of 0.4, with the default IoU threshold and input image size. Its `scannet200_classes.txt` vocabulary contains 192 distinct nonempty labels; `wall`, `floor`, `ceiling` are excluded as background. A pretrained Qwen3.5-4B, used without additional fine-tuning, ranks a sorted list of unique categories detected in past observations by their relevance to exploration. It receives the goal, including its image when applicable, but no archived scene images. We keep the first ten unique exact label matches. A frame is eligible if any nonbackground detection matches these labels. The unchanged retriever selects up to four eligible frames and orders them chronologically. If fewer than four frames remain, all are

used without adding extra images; if none remain, no memory images are provided. The archive itself is unchanged by filtering.

State reset. At each subtask transition, we reset both earlier-subtask memory access and exploration state. Goal-conditioned retrieval remains available over observations accumulated within the new subtask. Table 9 applies this joint reset to all three navigation policies.

Table 9: Effect of retaining memory and exploration state across subtasks on all GOAT-Bench Val-Unseen episodes. State reset clears earlier-subtask memory access and exploration state together. Decreases use the displayed one-decimal values.

MLLM backbone	Full		State reset		Decrease	
	SR	SPL	SR	SPL	SR	SPL
Qwen2.5-VL-3B	73.3	56.5	67.3	36.1	6.0	20.4
Qwen3-VL-4B	80.7	62.9	75.4	49.7	5.3	13.2
Qwen3.5-4B	81.0	63.2	78.6	53.4	2.4	9.8

B.4 Effect of the Number of Retrieved Memory Images

The main experiments use $K = 4$, chosen from training-data retrieval statistics. Table 10 reports the retrieval hit rate on the same original expert recall decisions retained across all tested values of K. The hit rate is the fraction of decisions with at least one retrieved historical frame captured before the current panorama that satisfied the target-visibility criterion in Equation 3.

For the navigation experiments, we train and evaluate a separate policy for each $K \in \{1, \ldots, 8\}$ using the corresponding number of retrieved memory images. The retrieval hit rate increases with K, whereas navigation performance does not improve monotonically. The highest observed navigation scores occur at $K = 6$ or $K = 7$, depending on the backbone and metric. With Qwen3.5-4B, $K = 6$ yields the highest SR (82.0), while $K = 7$ yields the highest SPL (64.8). In the measured cost runs, $K = 6$ uses approximately 20% more prompt tokens per decision than $K = 4$. We use $K = 4$ as the default setting to limit the number of input images.

Table 10: Effect of the number of retrieved memory images. Retrieval uses training decisions; navigation uses separately trained policies on all 360 Val-Unseen episodes. Bold identifies the adopted value of K, not the highest score.

K	Training hit rate (%)	Qwen2.5-VL-3B		Qwen3.5-4B	
		SR	SPL	SR	SPL
1	75.3	72.5	55.5	79.7	62.2
2	79.7	72.4	56.2	80.9	63.8
3	82.5	72.6	56.3	80.4	63.4
4	84.5	73.3	56.5	81.0	63.2
5	85.9	73.1	55.9	80.5	63.5
6	87.1	73.8	56.8	82.0	64.6
7	88.2	73.8	56.6	81.6	64.8
8	89.2	72.6	55.5	80.6	63.5

C Target Localization and Action Execution

Depth-based localization. For `FOUND`, the controller uses the depth and camera pose recorded with the selected image. The predicted box is rescaled to the depth resolution and clipped to image bounds; width or height below one depth pixel is invalid. We take the median finite depth within the central 40% of the box's width and height, requiring at least one valid sample. If none exists, we use the full box; if that also fails, localization is invalid. The depth is back-projected along the box-center ray and transformed using the camera pose recorded with that image. This applies to both

current and memory grounding. A nearby reachable destination is constructed from the resulting point; `EXPLORE` uses the representative navigable position of the selected frontier.

Destination selection. We project points around the estimated 3D target onto Habitat-Sim's global navmesh. Among reachable points with horizontal snapping displacement at most 0.15 m, we select the point closest in horizontal distance to the selected image's capture position. If none remains, we relax the displacement limit and choose the reachable snapped point with the shortest geodesic distance from the agent. If no reachable point exists or depth localization fails, we fall back to the image's capture position.

Execution and stopping. Each decision captures a panorama and normally makes one MLLM call. As in 3D-Mem (Yang et al., 2025) and AstraNav-Memory (Hu et al., 2026), we use Habitat-Sim's pathfinder (Savva et al., 2019; Szot et al., 2021; Puig et al., 2024) to plan shortest paths on the global navmesh. The controller follows these paths using discrete forward and turning actions. After reaching the destination selected by current-view grounding, the agent declares a stop. Memory grounding instead triggers a return visit, a new panorama, and another decision; recall requires current-view confirmation before stopping. Exploration is also followed by a new decision after movement. Invalid decisions fall back to an available frontier; stalled movement triggers replanning or unsuccessful termination. Budgets are 500 actions per subtask and 5,000 per episode, including panorama turns.

D Visual Grounding versus Coordinate Prediction

For each MLLM backbone, policies predicting bounding boxes or coordinates are separately trained on the same 52,961 decision-level examples with the same image-input preparation, retrieval, and optimization settings. The supervised output representation changes; all coordinate policies share coordinate labels. A predicted agent-relative destination in meters is transformed to world coordinates and projected onto navigable ground. Both variants retain simulator pose, NavMesh-based movement, the frontier interface, and confirmation of recalled targets in the current view. Table 11 reports the results.

Table 11: Output representation on all 360 GOAT-Bench Val-Unseen episodes (2,669 goals). Bounding-box grounding uses depth-based destination computation; coordinate prediction directly estimates an agent-relative destination.

MLLM backbone	Output and localization	SR	SPL
Qwen2.5-VL-3B	Box + depth	73.3	56.5
Qwen2.5-VL-3B	Direct coordinates	59.8	48.7
Qwen3-VL-4B	Box + depth	80.7	62.9
Qwen3-VL-4B	Direct coordinates	76.5	62.7
Qwen3.5-4B	Box + depth	**81.0**	63.2
Qwen3.5-4B	Direct coordinates	79.2	**65.4**

E Frontier Mapping and Baseline Comparisons

Main map. In the main setting, camera poses are used to update the observed region of a simulator-provided top-down navigability map at the episode's starting height. Frontiers separate observed and unobserved navigable space. Frontier regions are tracked, assigned representative navigable positions, and associated with captured views; frontier state persists across goals.

TSDF variant. The alternative fuses observed depth to build frontiers. At episode start, it uses one downward depth observation and information about navigable ground within 1 m of the initial position, then restores the standard camera. Subsequent frontier updates use the observed TSDF grid. The trained policy, observation storage rule, retriever, simulator pose, and movement controller remain unchanged. Released 3D-Mem and SCOPE configurations use 1280×1280 cameras, a 120° field of view, and a 30° downward tilt, versus RecallNav's level 720×640, 110° camera.

Evaluation. Table 12 reports results on all Val-Unseen episodes and on a subset containing the first episode of each scene (36 episodes, 278 goals). Values for snapshot-based baselines are taken from the original papers and retain their respective sensing and control configurations. On the whole set, TSDF lowers RecallNav's SR and SPL relative to its main map, while all three navigation policies remain above the reported snapshot-based baselines.

Table 12: Frontier variants on GOAT-Bench Val-Unseen. Subset: 36 episodes / 278 goals; whole set: 360 / 2,669. Snapshot-based baseline values are taken from the original papers.

Method	MLLM backbone	Frontier	Subset		Whole set	
			SR	SPL	SR	SPL
3D-Mem (Yang et al., 2025)	GPT-4o	TSDF	69.1	48.9	62.9	44.7
SCOPE (Wang et al., 2026a)	GPT-4o	TSDF	73.7	53.5	66.8	46.5
HGR (Chen et al., 2026)	GPT-4o	TSDF	72.4	56.2	64.1	50.1
RecallNav	Qwen2.5-VL-3B	TSDF	74.5	55.9	70.7	52.7
RecallNav	Qwen3-VL-4B	TSDF	78.1	60.0	77.1	59.5
RecallNav	Qwen3.5-4B	TSDF	**82.0**	**63.5**	76.6	58.6
RecallNav	Qwen2.5-VL-3B	NavMesh	78.8	61.0	73.3	56.5
RecallNav	Qwen3-VL-4B	NavMesh	78.8	61.9	80.7	62.9
RecallNav	Qwen3.5-4B	NavMesh	80.2	62.0	**81.0**	**63.2**

F Inference Time and Storage Requirements

Measurement setup. All methods use TSDF frontiers; AstraNav-Memory's frontier map is replaced with RecallNav's TSDF tracker and initialization. 3D-Mem and SCOPE use their released code and configurations with `gpt-4o-2024-11-20`; AstraNav-Memory uses its released Qwen2.5-VL-3B checkpoint. Each method runs four workers, each assigned one H200 and 16 CPU cores with common thread limits. Models are warmed up before timing. Values average over 1,427, 1,929, 632, 710, and 641 decisions, respectively, in table order.

Inference-time measurement. A decision may contain multiple MLLM calls. The MLLM column sums their durations, including the API round trip for GPT-4o and local generation for Qwen. *Pipeline* includes storing new observations, retrieval, input preparation, parsing, and path planning; *Total* is the sum of these two columns. Reported timing excludes TSDF fusion and frontier extraction, simulator rendering and motion, logging, saving images for inspection, and evaluation metrics. Total therefore reports the measured processing stages, not end-to-end runtime. Visual preprocessing is included in MLLM or pipeline depending on the implementation.

Storage measurement. Storage is measured as the size of retained raw data in decimal MB, excluding model weights, the simulator, and the TSDF map. *Frames* denotes the number of stored images, not the number supplied to the policy. *Images* contains RGB arrays; *Depth/3D* contains per-frame depth for RecallNav and object point clouds for 3D-Mem/SCOPE; *Other* contains descriptors, features, poses, and graph state. For each RecallNav frame, the measurement includes 720×640 RGB (1.38 MB), depth (0.92 MB), and a 2,048-dimensional descriptor (8 KB). 3D-Mem/SCOPE retain 1280×1280 RGB arrays, while AstraNav-Memory stores 720×640 images. The number of historical images supplied to the policy is limited, but storage grows with the episode.

G Evaluation Protocol

Evaluation splits. All GOAT-Bench evaluation scenes are excluded from policy training. Val-Seen, Val-Seen-Synonyms, and Val-Unseen contain 360/2,757, 340/2,525, and 360/2,669 episodes/goals. Goals are provided sequentially, and each subtask starts where the preceding one ends. LMEE-Bench evaluation uses all 166 released episodes and 405 questions; the released filter removes three descriptions longer than 55 words from 824 navigation goals, leaving 821.

Table 13: Per-decision inference time and storage on 36 GOAT-Bench Val-Unseen episodes (278 goals), using TSDF frontiers. Total = MLLM + pipeline under the stated exclusions; storage measures retained raw data, not total device memory. †GPT-4o latency includes the API round trip.

Method	MLLM backbone	Per decision			Latency per decision (s)			Stored data at decision (MB)			
		Tokens	Calls	Frames	MLLM	Pipeline	Total	Images	Depth/3D	Other	Total
3D-Mem (Yang et al., 2025)	GPT-4o†	4.6k	2.0	134	4.47	4.04	8.51	662	13.6	0.1	676
SCOPE (Wang et al., 2026a)	GPT-4o†	9.7k	3.9	159	9.25	4.30	13.55	784	15.5	0.2	800
AstraNav-Memory (Hu et al., 2026)	Qwen2.5-VL-3B	6.1k	1.0	229	1.25	0.55	1.80	317	–	<0.1	317
RecallNav (ours)	Qwen2.5-VL-3B	7.4k	1.0	389	1.41	1.28	2.69	538	359	3.5	900
RecallNav (ours)	Qwen3.5-4B	5.8k	1.0	385	1.51	1.23	2.74	532	355	3.4	890

Navigation metrics. RecallNav success requires a declared stop less than 1 m geodesically from a valid goal viewpoint. Category goals accept any matching instance; description and image goals require the designated instance. For success indicator S_i, traveled distance p_i, and shortest geodesic distance ℓ_i from the subtask start to a valid viewpoint, we compute SR and SPL as follows. Failures contribute zero. Navigation and QA scores are rounded to one decimal place.

$$\mathrm{SR} = \frac{100}{N}\sum_i S_i, \qquad \mathrm{SPL} = \frac{100}{N}\sum_i S_i \frac{\ell_i}{\max(p_i, \ell_i)}. \tag{4}$$

Sensing and actions. RecallNav uses a level camera at height 1.31 m with 720×640 RGB-D observations and a 110° horizontal field of view. Forward actions move 0.25 m and turns rotate 30°. Four panorama views are retained from twelve turns, all counted as actions. Agent radius is 0.17 m; sliding is disabled.

QA input and scoring. A pretrained Qwen2.5-VL-7B model, used without additional fine-tuning, generates a retrieval query for each question. Qwen3-VL-Embedding-2B retrieves from the navigation archive; the top three frames from the four-frame retrieval are displayed chronologically at 360×360. The same pretrained model answers from the question and these images, plus choices for multiple-choice QA, without an object list. Acc measures exact-match accuracy after label normalization, with answer positions balanced as described in Appendix I. For the open-ended QA metric Score, Qwen3-VL-30B-A3B-Instruct rates each answer under the LMEE-Bench rubric: $r \in \{1, \ldots, 5\}$ maps to $25(r-1)$, averaged over all 405 questions.

G.1 LMEE-Bench Baseline Re-evaluation

Navigation. We use the released LMEE-Bench evaluation code for 3D-Mem, RA-Mem, and MemoryExplorer on the same 166 episodes. Their shared configuration uses 1280×1280 images, a 120° field of view, 30° downward tilt, detector-based snapshots, TSDF frontiers, and Qwen2.5-VL-7B. Each subtask allows 50 decision steps. The released success metric uses the final position within 1 m geodesically of a valid viewpoint without requiring a declared stop. These are system-level comparisons with different sensing, control budgets, and stopping rules from RecallNav.

Configurations. MemoryExplorer uses its released checkpoint, exploration prompt, and controller. The release supplies no configurations for the other two baselines, so we configure them according to their descriptions. 3D-Mem keeps its original exploration prompt, parser, category pre-filter, model queries at each decision step, and mapping parameters, replacing GPT-4o with pretrained Qwen2.5-VL-7B-Instruct without additional fine-tuning. RA-Mem replaces the category pre-filter with a generated query matched by CLIP against snapshot images and object labels, taking three results from each, and queries the model again after reaching its selected target. Table 2 separates published and re-evaluated navigation results; RA-Mem's reported SPL is 11.4 versus 8.3 in our evaluation. The lower SPL indicates lower aggregate normalized path efficiency; differences in successful subtasks and their starting positions may also contribute.

Question answering. For Table 2, 3D-Mem, RA-Mem, and all RecallNav variants use pretrained Qwen2.5-VL-7B without additional fine-tuning; MemoryExplorer retains its released fine-tuned model. Each method uses its own accumulated memories and selection procedure. The baselines

retain the released QA prompt. All methods use the same reordered choices with balanced correct-answer positions (Appendix I) and the same evaluation model for open-ended answers. Missing answers count as incorrect and are rated as empty answers. These comparisons evaluate end-to-end memory, retrieval, and answering pipelines rather than memory quality in isolation.

H GOAT-BENCH RESULTS BY GOAL TYPE

Table 14 reports all three goal types on all episodes of each validation split under the main frontier setting. Each cell includes all goals of that type, including failures; all policies are evaluated on the same set of goals.

Table 14: GOAT-Bench SR/SPL by goal type for the three SFT policies. Category/Description/Image goal counts are 986/881/890 for Val-Seen, 881/847/797 for Val-Seen-Synonyms, and 991/856/822 for Val-Unseen. Values cover all episodes of the corresponding split and use the main NavMesh frontier setting.

Split	MLLM backbone	Category		Description		Image	
		SR	SPL	SR	SPL	SR	SPL
Val-Seen	Qwen2.5-VL-3B	68.4	50.3	65.0	50.1	79.9	61.8
Val-Seen	Qwen3-VL-4B	72.6	54.2	69.4	55.5	84.8	66.5
Val-Seen	Qwen3.5-4B	**73.3**	**54.9**	**70.8**	**55.8**	**86.0**	**68.5**
Val-Seen-Syn.	Qwen2.5-VL-3B	74.0	54.9	68.0	50.8	78.7	57.1
Val-Seen-Syn.	Qwen3-VL-4B	77.6	57.5	73.4	57.4	85.7	65.7
Val-Seen-Syn.	Qwen3.5-4B	**80.1**	**62.0**	**75.2**	**58.0**	**89.5**	**70.2**
Val-Unseen	Qwen2.5-VL-3B	74.7	56.9	64.1	49.2	81.3	63.4
Val-Unseen	Qwen3-VL-4B	81.1	62.5	**71.3**	**55.6**	89.9	71.0
Val-Unseen	Qwen3.5-4B	**82.6**	**63.8**	69.3	54.2	**91.1**	**71.8**

I SENSITIVITY TO ANSWER-CHOICE ORDER IN LMEE-BENCH

Balancing answer positions. In the released order, 247 of 405 correct answers occupy option B. With seed 42, we permute choices and remap labels, balancing positions separately for 66 two-choice questions (33/33) and 339 four-choice questions (85/85/85/84). Overall A/B/C/D counts become 118/118/85/84. All methods use the same permuted choice order for each question.

Paired evaluation. For each original/reordered comparison in Table 15, questions, retrieved images and their order, prompt, QA model, and decoding settings are fixed; only choice order and the corresponding answer labels change. The first six rows use RecallNav archives; the final three reuse each baseline's stored QA inputs. The main QA comparison uses the reordered choices with balanced answer positions for all methods.

Table 15: Sensitivity to answer-choice order on the same 405 questions. The two accuracy columns use identical questions and images; Balanced Acc uses reordered choices with balanced correct-answer positions. Baselines use their own memories; missing answers count as incorrect.

Memory source	QA model	Original Acc	Balanced Acc
RecallNav (Qwen2.5-VL-3B)	Qwen2.5-VL-7B	60.2	57.8
RecallNav (Qwen3-VL-4B)	Qwen2.5-VL-7B	61.7	58.3
RecallNav (Qwen3.5-4B)	Qwen2.5-VL-7B	61.5	57.0
RecallNav (Qwen2.5-VL-3B)	MemoryExplorer	64.4	54.3
RecallNav (Qwen3-VL-4B)	MemoryExplorer	67.4	58.3
RecallNav (Qwen3.5-4B)	MemoryExplorer	65.4	56.8
3D-Mem (baseline)	Qwen2.5-VL-7B	45.9	42.2
RA-Mem (baseline)	Qwen2.5-VL-7B	58.0	51.9
MemoryExplorer (baseline)	MemoryExplorer	66.7	53.6

J LMEE-BENCH RESULTS BY DIFFICULTY

Table 16 separates navigation goals and QA questions by difficulty. Counts therefore differ between the two tasks. Acc uses the same reordered choices with balanced answer positions; Score for open-ended QA is independent of choice order. QA models and baseline configurations follow Appendix G.1.

Table 16: LMEE-Bench results by difficulty. RecallNav uses $K = 4$ for navigation and three images for QA with the same pretrained Qwen2.5-VL-7B QA model, used without additional fine-tuning. Counts denote goals in (a) and questions in (b). Baselines use the same reordered choices and evaluation model for Score. Model names in RecallNav rows denote navigation backbones. QA models are specified in Appendix G.1.

(a) Multi-goal navigation

Method (MLLM backbone)	Easy (117)		Medium (559)		Hard (145)		Total (821)	
	SR	SPL	SR	SPL	SR	SPL	SR	SPL
3D-Mem (Yang et al., 2025)	31.6	13.7	17.2	6.6	14.5	5.3	18.8	7.4
RA-Mem (Wang et al., 2026b)	29.1	11.6	18.4	7.6	19.3	8.2	20.1	8.3
MemoryExplorer (Wang et al., 2026b)	29.9	20.0	23.1	13.7	19.3	10.9	23.4	14.1
RecallNav (Qwen2.5-VL-3B)	48.7	26.1	53.0	35.9	57.2	39.8	53.1	35.2
RecallNav (Qwen3-VL-4B)	52.1	**38.1**	57.4	40.8	**62.8**	**43.6**	57.6	40.9
RecallNav (Qwen3.5-4B)	**56.4**	36.7	**62.8**	**42.6**	58.6	40.6	**61.1**	**41.4**

(b) Memory-based question answering

Method / navigation policy	Easy (80)		Medium (271)		Hard (54)		Total (405)	
	Score	Acc	Score	Acc	Score	Acc	Score	Acc
3D-Mem (Yang et al., 2025)	33.1	33.8	38.6	42.8	33.3	51.9	36.8	42.2
RA-Mem (Wang et al., 2026b)	41.6	53.8	43.9	51.3	37.5	51.9	42.6	51.9
MemoryExplorer (Wang et al., 2026b)	42.8	48.8	45.2	52.8	35.6	64.8	43.5	53.6
RecallNav (Qwen2.5-VL-3B)	35.3	48.8	48.9	57.6	42.1	**72.2**	45.3	57.8
RecallNav (Qwen3-VL-4B)	**46.6**	**56.3**	50.1	**58.7**	**44.9**	59.3	**48.7**	**58.3**
RecallNav (Qwen3.5-4B)	35.6	48.8	**50.8**	58.3	42.1	63.0	46.7	57.0