SPARK: SELF-CORRECTING EMBODIED MEMORY VIA PHYSICAL RE-OBSERVATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Embodied retrieval-augmented generation (RAG) systems build semantic knowledge bases (KBs) from robot observations and answer language queries over them. Most retain and extend the first episode's KB, even though perception errors can introduce incorrect labels, merged instances, and missing regions from the outset. We present SPARK, a corrective multi-episode framework for embodied RAG. After the first episode, a semantic critic flags implausible and under-covered regions without ground truth, a targeted planner generates revisit viewpoints, and new observations are incorporated through a selective merge policy that can add, remove, or replace information. We evaluate SPARK on four outdoor AirSim maps under two viewpoint degradations using a perception-built KB, a visibility-matched development-time oracle, and a fixed EQA set scored before and after correction. One corrective episode improves downstream question answering on every map under both degradations while storing 19% fewer objects than append-only updating, with gains up to $2.6\times$ on a single map and condition. Extending correction through Episode 4 shows that the resulting KB can exceed the EQA performance of a single undegraded collection pass. Ablations show contributions from the critic, targeted revisits, and selective merge. SPARK shows that physical re-observation can correct persistent robotic memory by identifying weak regions, collecting new evidence, and selectively updating the KB.

1 INTRODUCTION

Retrieval-augmented generation (RAG) originated in large language model (LLM) research, where an external knowledge base (KB) is semantically searched to retrieve query-relevant information into the model's context (Lewis et al., 2020). This allows the model to draw on non-parametric knowledge beyond what is encoded in its weights, an idea that has evolved within NLP through variants such as graph-based RAG (Edge et al., 2024). More recently, RAG has extended into robotics, where embodied memory systems collect and organize multimodal observations into persistent representations, such as scene graphs or hierarchical semantic structures, that support retrieval, reasoning, and long-horizon embodied question answering (EQA) (Xie et al., 2024; Anwar et al., 2025).

Embodied memory, however, is constructed from noisy and incomplete physical observations. Perception can introduce incorrect objects, wrong semantic labels, inaccurate depth estimates, and incomplete observations that persist in the KB and propagate to downstream tasks long after the original observation was made. This raises a fundamental question: *how can a robot detect and correct errors in a memory that has already been built?* In NLP, unreliable retrieved knowledge has motivated corrective mechanisms such as CRAG and Self-RAG (Yan et al., 2024; Asai et al., 2024). These approaches, however, operate on retrieval or generation within a language model and cannot return to the physical world for new evidence.

Correcting an already-built embodied memory requires both identifying unreliable stored knowledge and acquiring new evidence to resolve it. Existing systems, however, typically address these capabilities separately. One line of work, often built on dense photometric representations such as point clouds or 3D Gaussians, treats stored information as correctable through reinterpreting observations, retrieving previously captured frames, or synthesizing views implicit in the reconstruction, and so cannot recover information that was never adequately observed (Chang et al., 2026; Zhu et al., 2026; Ali et al., 2025; Lu et al., 2026). Another uses physical interaction or re-observation to

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.

acquire new information without correcting an existing memory: Mind Palace interprets discrepancies with past memory as changes in the environment (Ginting et al., 2025), while REFLECT uses prior experience to recover from an immediate task failure (Liu et al., 2023). What remains missing is a system that diagnoses an already-built KB independently of any particular query, and resolves what it finds through newly acquired physical evidence.

To bridge this gap, we present SPARK, short for *Self-Corrective Process for Active Robotic Knowledge*. A VLM-powered *critic* identifies suspect regions in the KB, independently of any particular EQA query, and a planner converts those diagnoses into targeted revisits where the robot acquires new observations. This evidence is then selectively merged into the KB, closing the loop between diagnosis, re-observation, and correction across successive episodes.

We implement SPARK with an RGB-D perception stack in AirSim (Shah et al., 2018), combining open-vocabulary segmentation, depth backprojection, and VLM-based captioning. We evaluate EQA after correction and separately assess the critic against a visibility-matched oracle graph available only during development. The oracle follows the same waypoints and viewing constraints as the perception pipeline. We validate SPARK across four simulated scenes, multiple initial viewpoint configurations, successive corrective episodes, and ablations of the main loop components: merge policy, critic mode, and revisit policy.

Our **contributions** are:

- SPARK, to our knowledge the first framework where a robot repairs its own knowledge base without ground truth, using a semantic critic to diagnose memory, a planner to target revisits, and a selective corrective merge that removes and supersedes stored content. Correction improves EQA on every map and starting configuration, and stores 19% fewer objects than appending the same observations.
- An evaluation setting for memory correction, pairing a realistic perception stack with a development-time oracle restricted to the same viewpoints and visibility, and a fixed EQA set of 151 questions across four outdoor AirSim maps. Critic-guided revisits beat an untargeted baseline under a matched budget, and by the fourth episode every map exceeds what a robot recovers from the undegraded viewpoints.

2 Related Work

Building Embodied Memory. Robots must often use past observations for later tasks. Memory has traditionally relied on metric-dense and topological maps that preserve observations as persistent spatial representations (Henriques & Vedaldi, 2018), but as robots increasingly act on natural-language instructions (Ahn et al., 2023; Driess et al., 2023; Huang et al., 2023), memory has shifted toward representations that are semantic and language-grounded as well as spatial. ReMEmbR (Anwar et al., 2025) stores captioned observation segments in a flat vector database, while Embodied-RAG (Xie et al., 2024) organizes them by spatial-semantic clustering into a hierarchy that supports retrieval and region-level updates, which we build on. ConceptGraphs instead lifts an existing dense geometric map into an open-vocabulary scene graph (Gu et al., 2024). Across these approaches, memory quality depends on what was encoded during observation, and little attention has been paid to correcting information that was stored incorrectly.

Corrections in Memory Systems. The idea of memory correction was first articulated in NLP at different stages of RAG. CRAG treats retrieved evidence as unreliable and corrects it before use (Yan et al., 2024), while Self-RAG learns when to retrieve and to critique its own output against retrieved evidence (Asai et al., 2024). Neither operates on persistent memory built from physical observations or can gather new evidence by re-observing the world. In robotics, Mind Palace structures long-term deployments into a sequence of scene-graph world instances and can explore the current environment to answer a query (Ginting et al., 2025). It interprets discrepancies with past instances as environmental changes and preserves them as history. REFLECT uses a hierarchical summary to diagnose task failure and plan a corrective action (Liu et al., 2023), with correction applied to the task instead of persistent memory. SPARK instead studies how an already-built embodied memory carrying imperfect information can be corrected to improve downstream EQA.

Editable Scene Representations. A related line of work edits existing scene representations using previously collected evidence. RAG-3DSG rectifies low-confidence captions using synthetic views from the reconstructed point cloud and neighboring high-confidence nodes (Chang et al., 2026), while OGScene3D reassigns low-confidence clusters from nearby high-confidence Gaussians (Zhu et al., 2026). Both therefore infer corrections from information already encoded, without acquiring new physical evidence for under-observed regions. GraphPad retrieves additional keyframes from an existing scan when the current scene graph cannot answer a query (Ali et al., 2025), limiting its updates to previously captured information and making them query-driven. GSMem shares our premise that a region may be under-observed, but triggers physical exploration only when a query cannot be answered from a rendered view of its Gaussian-splatting map (Lu et al., 2026), making exploration a query-time fallback rather than a standing diagnosis of the KB. SPARK diagnoses regions independently of downstream queries and physically revisits those it flags, allowing it to recover information that the original observations never captured adequately.

3 METHOD

3.1 PROBLEM FORMULATION

Current persistent memory systems generally assume a single collection pass suffices for downstream EQA, yet a KB built from a real perception stack can accumulate errors—missing objects, incorrect labels, misplaced geometry—that propagate to downstream tasks unless corrected. We study KB correction over repeated episodes in a static world, where cross-episode mismatches reflect memory errors as opposed to scene changes.

An episode is a single traversal of a set of N_e waypoints, with one observation recorded per waypoint, so coverage is determined by the waypoints without continuous recording.

During episode e, the robot records $\mathcal{O}^{(e)} = \{\, o_{e,k} \,\}_{k=1}^{N_e}$, where $o_{e,k} = \big(I_{e,k}, D_{e,k}, \mathbf{p}_{e,k}\big)$: I is the RGB image, D the depth map, and $\mathbf{p}$ the robot pose. Pose is taken as ground truth because localization is not the subject of this work. The first episode builds the memory

$$H^{(1)} = f_{\text{build}}\big(\mathcal{O}^{(1)}\big), \tag{1}$$

by pooling and clustering observations; thus, visit order does not affect the result.

Given $H^{(e)}$, a semantic critic $\mathcal{C}$ inspects it together with the per-episode evidence $Z^{(e)}$ retained alongside it—the waypoint images and their provenance—and returns regions flagged as unreliable,

$$\mathcal{R}^{(e)} = \mathcal{C}\big(H^{(e)}, Z^{(e)}\big). \tag{2}$$

A revisit planner converts these regions into waypoints, $\mathcal{V}^{(e+1)} = f_{\text{plan}}(\mathcal{R}^{(e)}, H^{(e)})$, and visiting them yields targeted observations $\mathcal{O}^{(e+1)}$. The memory is then corrected as

$$H^{(e+1)} = f_{\text{correct}}\big(H^{(e)}, \mathcal{O}^{(e+1)}\big). \tag{3}$$

The formulation assumes no particular memory representation—only a persistent KB, a critic, a planner, and a corrective merge that can remove as well as add information. The objective is for $H^{(e+1)}$ to answer questions about the environment better than $H^{(e)}$. We evaluate the first corrective transition, $H^{(1)}$ to $H^{(2)}$, and study repeated correction separately in Sec. 6.5. Ground truth comes from a development-time oracle used only for scoring and never available to the deployed system. Figure 1 summarizes this correction loop.

3.2 PERCEPTION-BASED MEMORY CONSTRUCTION

We keep the hierarchical memory representation introduced by Embodied-RAG (Xie et al., 2024) and illustrated in Figure 2, since its hierarchy allows corrective updates to act directly on regions flagged by the critic. Its original collection process, however, populates memory from oracle simulator state, so every stored object exists and is correctly labeled and positioned. Our setting requires a KB that can actually be wrong, so we instead build memory from the robot's own sensors: the pipeline receives only synchronized RGB $I_{e,k}$ and depth $D_{e,k}$ at each waypoint, with no simulator identities, object poses, or visibility annotations.

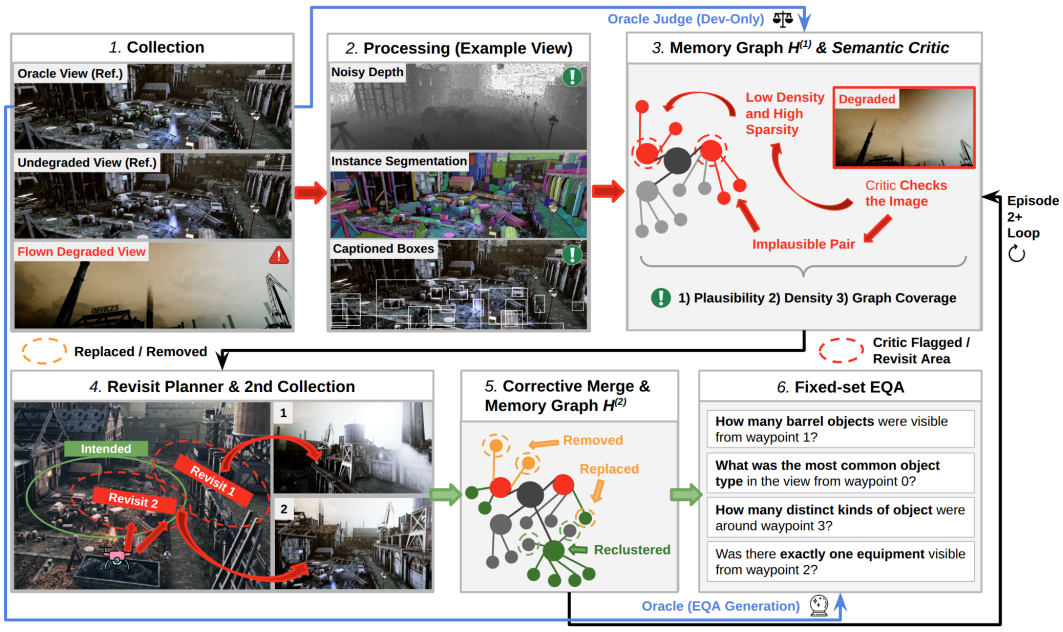


Figure 1: **The SPARK correction loop.** Episode 1 flies the source waypoints under a degraded view and builds memory $H^{(1)}$ from its own RGB-D observations (1–2); panel 2 uses an example pose for legibility. The semantic critic reads $H^{(1)}$ and the Episode 1 images alone, flagging suspect regions by plausibility, density, and graph-support coverage (3). The planner turns those into source-anchored viewpoints for Episode 2 (4), and the corrective merge removes, replaces, and adds entries to produce $H^{(2)}$ (5), which a fixed EQA set scores against $H^{(1)}$ (6). The oracle (blue) is development-time only and never reaches the critic, planner, or merge.

A segmentation model, SAM2 (Ravi et al., 2025), separates each frame into candidate object instances. We lower the segmenter's confidence threshold to favor over-detection, since a spurious mask can be rejected downstream whereas an object it never proposes cannot be recovered. We filter and group its masks before passing candidates to the VLM. Each retained mask is then presented to the VLM in three views, a tight crop of the instance, the same crop with its background suppressed, and the full scene with the instance outlined, which lets the model use surrounding context when determining what the region contains, and returns a label for valid objects while rejecting background regions. We localize each object from its mask and a high depth quantile then backproject it to world coordinates using the vehicle pose.

To avoid unrealistically perfect simulated depth, we apply sensor-realistic depth noise while leaving pose unperturbed, since pose is the controlled variable in this study. The resulting object nodes are pooled and clustered into the initial hierarchical memory $H^{(1)}$, retaining realistic segmentation, labeling, and localization errors for the subsequent method to identify and repair. Clustering combines spatial proximity with semantic similarity computed using `text-embedding-3-small` (Neelakantan et al., 2022) and a co-visibility constraint requiring at least one shared observing waypoint, preventing proximity alone from merging objects across occlusions or opposite approaches.

3.3 Semantic Critic

After Episode 1, the robot must diagnose errors from its own memory and observations, $\mathcal{R}^{(1)} = \mathcal{C}\big(H^{(1)}, Z^{(1)}\big)$. The critic is a scaffolded VLM based on `gpt-5-mini` (Singh et al., 2025) that uses graph context and the supporting images $Z^{(1)}$ to identify potential errors in $H^{(1)}$, without access to oracle data or external references.

The critic uses three complementary signals: **plausibility**, which tests whether stored labels are visually and contextually supported; **density**, which compares visible object estimates with stored

objects per waypoint; and **graph-support coverage**, which identifies clusters supported by too few observations or nodes. Together, these capture incorrect labels, under-recovered regions, and weakly observed parts of the graph. The critic returns plausibility scores and suspect waypoints and clusters, defining the regions $\mathcal{R}^{(1)}$ for Episode 2 to revisit. Thresholds are given in Appendix C.

3.4 Corrective Episode and Merge

The flagged regions from diagnosis must be converted into revisit viewpoints that the robot can physically execute, $\mathcal{V}^{(2)} = f_{\text{plan}}(\mathcal{R}^{(1)}, H^{(1)})$.

Revisit planner. Nearby flagged clusters from the same source waypoint are merged into a single revisit region. For each region, the planner generates fixed forward, backward, and lateral offsets from the source waypoint, all aimed at the region centroid. Anchoring revisits to the original viewpoint avoids relying on uncertain region geometry and isolates targeted revisitation from unconstrained viewpoint search. Episode 2 otherwise uses the same motion and recovery procedure as Episode 1. Implementation details are given in Appendix D.

Corrective merge. Episode 2 observations pass through the same perception pipeline and are selectively merged into $H^{(1)}$, thus $H^{(2)} = f_{\text{correct}}\big(H^{(1)}, \mathcal{O}^{(2)}\big)$. Within flagged regions, supported observations may add, remove, or relabel nodes, while unsupported singletons are discarded. Outside $\mathcal{R}^{(1)}$, $H^{(1)}$ remains unchanged, isolating correction to regions identified by the critic.

4 Development-Time Oracle

Evaluating whether correction improves memory requires a reference for what was actually observable, which is impractical to verify manually at scale. We therefore construct a development-time oracle from simulator identities and positions, restricted to the same waypoint poses, field of view, range, and geometric visibility as the perception pipeline. The resulting graph H^{oracle} is used only for evaluation and is never available to the critic, planner, or corrective merge. The oracle is not a canonical description of the scene—simulator asset names often differ from natural object labels, and visible assets include incidental elements such as decals, debris, and lighting props—so we evaluate $H^{(1)}$ against H^{oracle} through structural and approximate semantic correspondence.

Oracle judge. The oracle judge provides a common evaluation layer between the perception-built memory and H^{oracle}. Because the two graphs may differ in segmentation, localization, and vocabulary, it compares spatial regions in place of requiring one-to-one object matches, using approximate semantic correspondence for labels. Within each region, it evaluates memory quality from object recovery, spatial distribution, and category agreement. It separately evaluates whether $\mathcal{R}^{(1)}$ identifies poorly recovered regions, reporting region-level precision, recall, and F1 for the critic and for each of its plausibility, density, and coverage signals. The same oracle regions provide ground-truth answers for the fixed EQA set.

5 Experiments and Evaluation

This section defines the experimental setting, evaluation protocol, metrics, and controls used to measure correction.

5.1 Simulation Platform and Scope

All experiments run in AirSim (Shah et al., 2018) on adapted Unreal Engine 4 environments using the containerized image from Embodied-RAG (Xie et al., 2024). Unlike Embodied-RAG, which constructs memory from simulator state, our perception pipeline builds memory from RGB-D observations while a separate oracle pipeline reads simulator state only for evaluation, both at 1280×720 resolution and 90° horizontal field of view. The study isolates perception and memory correction by treating *(i)* pose as exact, excluding localization error; *(ii)* simulator depth as exact before the sensor model is applied, making that model the controlled source of range error; and *(iii)* the environment

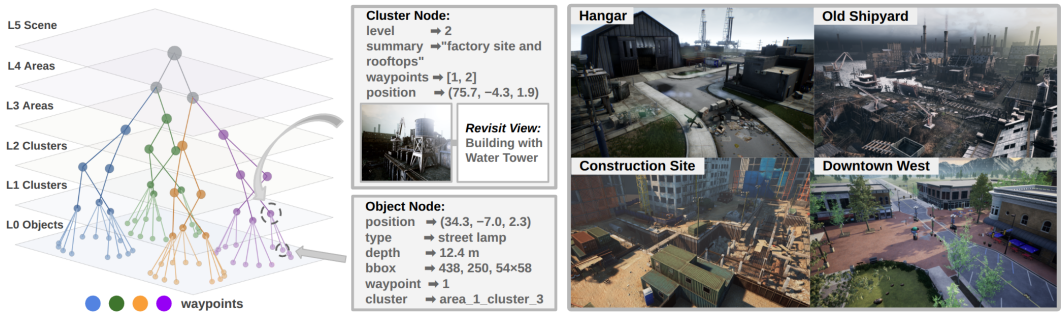


Figure 2: **Hierarchical memory and evaluation environments.** Left: objects are grouped into clusters and clusters into areas, and retrieval descends this hierarchy. A corrective episode acts at the object tier, removing entries Episode 2 fails to support and adding ones it newly recovers, after which the affected clusters are rebuilt. Right: the four outdoor maps spanning an enclosed courtyard, an open industrial yard, a construction area, and a bright suburban block.

as static within each correction window. Evaluation is restricted to outdoor environments because targeted correction requires regions that remain only partially observed from individual viewpoints, and Episode 2 revisit poses must be reachable without an occupancy map or obstacle-avoidance stack, which are outside the present scope.

5.2 Maps, Viewpoints, and Viewpoint Conditions

The evaluation uses the four outdoor maps shown in Figure 2: Hangar, Old Shipyard, Construction Site, and Downtown West. Each map has four hand-selected source waypoints chosen for object diversity, density, and regional isolation, and Episode 1 follows the same fixed source locations in every run. One observation is recorded at each waypoint and none in transit.

Each map is evaluated under one baseline and two degraded viewpoint conditions. *Baseline* uses the source poses unchanged; *pitch-up* tilts the camera upward, removing object-bearing near-field content; and *low-lateral* offsets the camera laterally and downward. The oracle always observes the baseline poses, including when the perception run is degraded, so the oracle graph describes what is visible from a well-chosen view while the degraded run represents what the perception stack recovers from a poor one. Exact offsets are given in Appendix D.

5.3 Metrics

Critic and memory quality are scored by the oracle judge described in Sec. 4, with metric definitions and critic thresholds given in Appendix C. Our primary metric is fixed-set EQA, which evaluates $H^{(1)}$ and $H^{(2)}$ on identical questions to measure how correction changes the KB. Each question is tied to a source waypoint using stored provenance, preventing observations from other viewpoints from masking information lost at that waypoint. The question set covers four families—inventory, dominant type, type richness, and rare presence—to test whether correction recovers different forms of scene information.

Ground-truth answers come from the oracle's per-waypoint observations, and both KBs are queried through the same coarse vocabulary. A scaffolded LLM judge (`gpt-5-mini`) scores answers as correct (1.0), partial (0.5), or wrong (0.0), with adjacent answer bands receiving partial credit. Unscorable questions are excluded and reported separately. Question-family definitions, answer bands, and the full judging rubric are given in Appendix C.

5.4 Controls and Ablations

The main Episode 1-to-Episode 2 evaluation includes three ablations that isolate each component of SPARK: *(i)* the update ablation compares unchanged $H^{(1)}$, append-only updating, cluster-wide replacement, and the selective corrective merge; *(ii)* the critic ablation evaluates plausibility, density,

and coverage individually and together; and *(iii)* the revisit ablation compares critic-guided and untargeted placement under the same Episode 1 memory, observation budget, and merge. All four maps are evaluated, with Downtown West reported separately because its degraded viewpoint does not reduce EQA and reliable flight needs a ghost-transit workaround that would otherwise confound the comparison. Unreachable revisit candidates are skipped and reported so failed proposals remain part of the planner evaluation. Selective-merge support thresholds are given in Appendix A.

A separate longitudinal study follows correction through Episode 4 on all four maps using one viewpoint condition per map. Ablation controls are task-matched to the components tested. CRAG is not directly comparable because it corrects retrieval using alternative text evidence, not physical recollection; Mind Palace, without its question-conditioned planner and temporal objective, reduces to repeated sensing with append-only storage, which is already represented by the update ablation.

6 RESULTS

This section shows that growing, imperfect memory from realistic episodes can be improved by diagnosis, targeted re-observation, and selective correction, rather than append-only accumulation or indiscriminate replacement. We first report the Episode 1 to Episode 2 transition across four maps and two degraded viewpoint conditions, then use the four-map ablation set to isolate the merge policy, the critic mode, and the revisit policy. Lastly we perform an exploratory longitudinal study of how SPARK adapts to Episode 3 and Episode 4. Each EQA comparison uses the same fixed question set before and after the relevant intervention.

6.1 MAIN RESULTS: EPISODE 1 TO EPISODE 2

Table 1: EQA before and after one corrective episode, averaged over all four maps. The undegraded reference uses the same perception pipeline from the original source viewpoints.

Viewpoint condition	$H^{(1)}$	$H^{(2)}$	Δ
Pitch-up	0.324	0.458	+0.135
Low-lateral	0.408	0.498	+0.089
Overall	0.366	0.478	+0.112
Undegraded ref.	*0.515*		—

Table 2: Merge ablation over all four maps. All active rules use the same critic-guided Episode 2 observations.

Merge policy	Δ EQA	Objects
Unchanged $H^{(1)}$	+0.000	—
Append only	+0.082	209
Replace clusters	+0.063	175
Selective merge	+0.080	170

Correction improves EQA across all four maps and both viewpoint conditions, with larger gains for pitch-up, where Episode 1 starts from a lower score. Overall, one corrective episode raises EQA from 0.366 to 0.478, recovering much of the gap to the 0.515 undegraded reference despite revisiting only flagged regions. Downtown West behaves differently from the other three maps because its degraded views do not reduce EQA: its Episode 1 pitch-up memory scores 0.596 compared with 0.575 from the undegraded viewpoints, as the upward view still captures informative building facades. Correction still improves it, by $+0.054$ and $+0.104$, but from an undamaged starting point, so including it raises the reported $H^{(1)}$ averages and makes the overall gain a conservative estimate.

How much correction helps depends on what the first pass missed. Recovering missing object types can restore information needed for EQA, while adding more instances of types already present may expand the KB without changing its answers. This helps explain variation across maps and also why the size of the KB update does not directly predict the downstream gain.

6.2 MERGE ABLATION

Table 2 shows that the main effect is whether Episode 2 evidence updates the KB at all: all three active rules improve EQA within 0.019 of each other, but differ in what they carry forward. Append can only add, replacement rewrites the targeted cluster, and the selective merge retains, removes, or adds individual nodes—producing the smallest KB (170 objects vs. 209 for append, 175 for replacement) while remaining useful as evidence accumulates: append keeps prior errors as the KB

grows, and replacement can discard valid observations on revisit. A single transition gives these effects little time to diverge, consistent with the similar EQA gains observed here.

6.3 Critic Ablation

The corrective rule can repair only the regions that the critic identifies, so the next question is whether the three diagnostic signals provide genuinely complementary coverage. We run the critic using plausibility, density, and graph-support coverage individually, then run all three together, and score the flagged set against regions that the oracle shows to be poorly recovered.

Table 3: Critic ablation over all four maps, reporting region-level recall with true positives and false negatives.

Diagnosis	Recall	TP	FN
Plausibility only	0.306	15	34
Density only	0.286	14	35
Coverage only	0.531	26	23
All three	**0.694**	34	15

Table 4: EQA gain for critic-guided and untargeted revisitation under the same observation budget.

Map	Critic-guided	Untargeted
Old Shipyard	$\mathbf{+0.194}$	$+0.151$
Hangar	$\mathbf{+0.146}$	$+0.083$
Construction Site	$\mathbf{+0.167}$	$+0.038$
Mean (degraded)	$\mathbf{+0.169}$	$+0.091$
Downtown West	$\mathbf{-0.004}$	-0.041

Table 3 shows that the combined critic recovers more poorly reconstructed regions than any signal alone, both overall and on each map. Coverage performs best individually because it can identify regions with little supporting evidence, while plausibility and density are comparable, but combining all three raises recall from 0.531 to 0.694, confirming that the signals capture different failures.

Precision is omitted because every assessed region in these degraded memories falls below the coverage threshold, leaving no negative examples under this scoring rule. This would make precision 1.0 by construction and would not show that the critic is free of false positives. Precision becomes informative again after correction restores adequately recovered regions.

6.4 Revisit Ablation

After diagnosis, the system must decide where to spend its revisit budget. We compare the critic-guided planner with an untargeted baseline that samples clusters uniformly from the memory graph, using the same Episode 1 memory, observation budget, and corrective merge.

Table 4 shows that critic-guided revisitation improves EQA more on all three maps where the initial degradation reduced performance, with a mean gain of $+0.169$ compared with $+0.091$ for untargeted revisitation. The untargeted baseline still improves because the critic flags much of these scenes, making a random revisit likely to reach a region that needs correction. Targeting becomes more valuable as the flagged regions become more concentrated.

On Downtown West, the critic-guided policy makes almost no change (-0.004) because the initial degradation leaves the KB largely intact, while untargeted revisitation reduces EQA by 0.041. The revisit policy therefore allocates observations where correction is useful and avoids unnecessary updates when there is little to repair.

6.5 Longitudinal Study: Episode 1 to Episode 4

We extend the correction loop to Episode 4 on all four maps, with each episode diagnosing and revisiting the memory produced by the previous one. The study uses one viewpoint condition per map, with low-lateral used throughout because its Episode 1 performance is closer to the undegraded reference than pitch-up.

Figure 3 shows that all four maps eventually exceed their undegraded reference because that reference is still a single perception pass from one pose per waypoint. Additional views can recover objects that were occluded, merged, or poorly observed even from the original viewpoints. On the three maps where degradation initially reduces EQA, most of the improvement arrives in the first

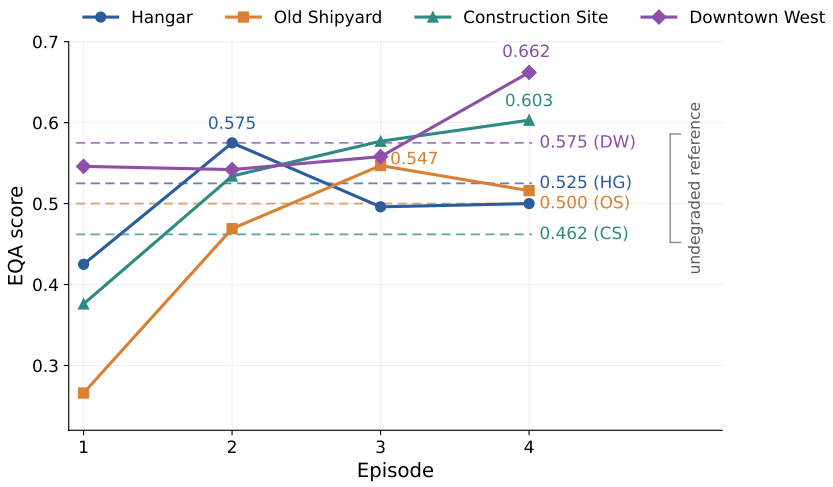


Figure 3: Fixed-set EQA across four episodes, with each map's undegraded reference as a dashed line. All maps eventually exceed their reference, though timing varies by map.

corrective episode and later gains are smaller. Downtown West starts near its reference and changes little through the first two corrections before improving by $+0.100$ at Episode 4, showing that the useful number of revisits depends on how much recoverable information remains.

Repeated correction increases the size and cost of the KB. Old Shipyard stores 1,732 objects for a scene with 152 oracle objects, even though each corrective episode is capped at thirty viewpoints. Sustaining the loop requires deduplication and cross-episode object identity so repeated observations can be associated with objects. Our initial merging used a type-scaled distance with a co-visibility requirement and still reduced EQA along with KB size, suggesting that proximity and shared observation alone are insufficient for cross-episode identity.

7 Conclusion

We presented SPARK, a corrective memory loop in which a robot diagnoses its KB without ground truth, revisits flagged regions, and selectively adds, removes, or updates what it stores. Across four outdoor scenes, a corrective episode improves downstream EQA when Episode 1 misses information, with critic-guided revisitation outperforming an untargeted policy under the same budget. When Episode 1 captures the scene well, the critic avoids unnecessary revisits, making targeted correction useful for both repairing memory and deciding when further observation is needed.

Our experiments isolate memory correction in a static world with known pose and a fixed motion routine. Longer deployments will require persistent object association to merge repeated observations and control KB growth. The same capability could help distinguish scene changes from perception errors and account for pose drift under real SLAM. Extending correction online during a mission and testing across indoor environments and other platforms would further expose different scene geometry and sensing constraints. SPARK demonstrates that a robot can use its own observations to identify weak parts of its KB, revisit them, and improve what it has stored.

AI use statement

We used generative AI tools to assist with language editing, including polishing, shortening, and clarifying prose and LaTeX formatting. We did not use generative AI for research ideation, experimental design, or interpretation of results. All technical content, claims, and conclusions were produced by the authors. We take responsibility for the final content of this work.

REFERENCES

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 287–318. PMLR, 2023.

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. Presented at the CVPR 2025 Workshop on 3D-LLM/VLA: Bridging Language, Vision and Action in 3D Environments.

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. doi: 10.1109/ICRA55743.2025.11127706.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In *The Twelfth International Conference on Learning Representations*, 2024.

Yue Chang, Rufeng Chen, Zhaofan Zhang, Yi Chen, Yifan Tian, and Sihong Xie. RAG-3DSG: Enhancing 3D Scene Graphs with Re-Shot Guided Retrieval-Augmented Generation. In *European Conference on Computer Vision (ECCV)*, 2026.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. PaLM-E: An Embodied Multimodal Language Model. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 8469–8488. PMLR, 2023.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From Local to Global: A Graph RAG Approach to Query-Focused Summarization. arXiv preprint arXiv:2404.16130, 2024.

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 *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 5072–5106. PMLR, 2025.

Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. 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. doi: 10.1109/ICRA57147.2024.10610243.

João F. Henriques and Andrea Vedaldi. MapNet: An Allocentric Spatial Memory for Mapping Environments. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8476–8484. IEEE, 2018. doi: 10.1109/CVPR.2018.00884.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Noah Brown, Tomas Jackson, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. Inner Monologue: Embodied Reasoning through Planning with Language Models. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 1769–1782. PMLR, 2023.

Kourosh Khoshelham and Sander Oude Elberink. Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications. *Sensors*, 12(2):1437–1454, 2012. doi: 10.3390/s120201437.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc., 2020.

Zeyi Liu, Arpit Bahety, and Shuran Song. REFLECT: Summarizing Robot Experiences for Failure Explanation and Correction. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 3468–3484. PMLR, 2023.

Yiren Lu, Yi Du, Disheng Liu, Yunlai Zhou, Chen Wang, and Yu Yin. GSMem: 3D Gaussian Splatting as Persistent Spatial Memory for Zero-Shot Embodied Exploration and Reasoning. arXiv preprint arXiv:2603.19137, 2026.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David Schnurr, Felipe Petroski Such, Kenny Hsu, Madeleine Thompson, Tabarak Khan, Toki Sherbakov, Joanne Jang, Peter Welinder, and Lilian Weng. Text and Code Embeddings by Contrastive Pre-Training, 2022.

Chuong V. Nguyen, Shahram Izadi, and David Lovell. Modeling Kinect Sensor Noise for Improved 3D Reconstruction and Tracking. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pp. 524–530. IEEE, 2012. doi: 10.1109/3DIMPVT.2012.84.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment Anything in Images and Videos. In *The Thirteenth International Conference on Learning Representations*, 2025.

Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. In Marco Hutter and Roland Siegwart (eds.), *Field and Service Robotics*, volume 5 of *Springer Proceedings in Advanced Robotics*, pp. 621–635. Springer International Publishing, 2018. doi: 10.1007/978-3-319-67361-5_40.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, et al. OpenAI GPT-5 System Card, 2025.

Quanting Xie, So Yeon Min, Pengliang Ji, Yue Yang, Tianyi Zhang, Kedi Xu, Aarav Bajaj, Ruslan Salakhutdinov, Matthew Johnson-Roberson, and Yonatan Bisk. Embodied-RAG: General Non-parametric Embodied Memory for Retrieval and Generation. arXiv preprint arXiv:2409.18313, 2024.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. Corrective Retrieval Augmented Generation. arXiv preprint arXiv:2401.15884, 2024.

Siting Zhu, Ziyun Lu, Guangming Wang, Chenguang Huang, Yongbo Chen, I-Ming Chen, Wolfram Burgard, and Hesheng Wang. OGScene3D: Incremental Open-Vocabulary 3D Gaussian Scene Graph Mapping for Scene Understanding. arXiv preprint arXiv:2603.16301, 2026.

Algorithm 1 SPARK correction loop, Part 1: Episode 1 collection and memory construction.

Require: waypoints $W = \{w_1, \ldots, w_4\}$, viewpoint condition c, depth envelope $R = 40$ m
Ensure: corrected memory $H^{(2)}$, EQA scores on $H^{(1)}$ and $H^{(2)}$

STAGE 1: EPISODE 1 COLLECTION

$P \leftarrow$ APPLYCONDITION(W, c) ▷ baseline, pitch-up ($+40.7^\circ$), or low-lateral
for each pose $p \in P$ **do**
  climb to 20 m, translate, descend, set yaw on airframe and pitch on gimbal, stabilize
  record $(I_t, D_t, \mathbf{p}_t)$ ▷ one frame per waypoint; nothing recorded in transit
end for

STAGE 2: BUILD $H^{(1)}$ FROM RGB-D ONLY

for each observation t **do**
  $M \leftarrow$ SEGMENT(I_t) ▷ IoU 0.50, stability 0.80, 32×32 grid
  discard masks with area < 300 px^2 or dimension < 20 px
  $D_t \leftarrow$ ADDSENSORNOISE(D_t) ▷ $\sigma_z = 0.002z^2$, 1.2 px jitter, 2% dropout on grazing surfaces
  $z_m \leftarrow Q_{75}$ of D_t over each mask m; discard if $z_m > R$
  $C \leftarrow$ 15 largest masks $\cup$ up to 5 groups of small masks clustered at 6% of the image diagonal
  for each candidate $m \in C$ **do**
    $\ell_m \leftarrow$ CAPTION(crop, masked crop, outlined frame)
    reject if ℓ_m is a non-object label, matches a background term, or fails person verification
    backproject z_m through the mask centroid to world coordinates
  end for
end for
pool all objects; cluster by complete linkage at level-ℓ threshold $5.0(1 + \ell)$ m, blocking merges that violate the semantic or co-visibility constraint
$H^{(1)} \leftarrow$ resulting hierarchy; $Z^{(1)} \leftarrow$ retained images and regional summaries

A CORRECTION LOOP

This appendix presents the full loop and the thresholds used at each stage, so the path from a captured frame to a corrected memory can be followed end to end. Algorithms 1–3 split the procedure into three parts: Episode 1 collection and memory construction, diagnosis and revisit planning, and the corrective episode with its development-time evaluation. The procedure is the same across viewpoint conditions, with the condition affecting only the poses produced by APPLYCONDITION in Stage 1 and remaining hidden from all decision-making components.

Evaluation remains separate from correction. The critic at Stage 3 reads only $H^{(1)}$ and $Z^{(1)}$, so no oracle information can affect $H^{(2)}$. The oracle first appears at Stage 6, after the corrected memory is fixed. At Stage 4, the critic determines which region to revisit, while the viewpoint is generated from the source waypoint and region geometry. A revisit can therefore still be useful when the critic identifies an unreliable region but misidentifies the underlying failure.

B PERCEPTION PIPELINE

B.1 SEGMENTATION AND CANDIDATE SELECTION

Masks are produced by SAM2 (Ravi et al., 2025), $\mathcal{M}_{e,k} = \mathrm{SAM}(I_{e,k})$, using predicted-IoU and stability thresholds below their defaults with a 32×32 point grid. These settings favor proposal recall because spurious masks can be rejected downstream, while objects the segmenter never proposes cannot be recovered. Candidates are filtered in the order given in Algorithm 1. Small masks are removed as likely texture fragments, masks beyond the sensing envelope are discarded after depth estimation, and the remaining masks are ranked for the captioning budget. The budget retains the 15 largest masks per waypoint and up to five spatial groups of smaller masks, formed at 6% of the image diagonal. We write $\widehat{\mathcal{M}}_{e,k} \subseteq \mathcal{M}_{e,k}$ for the retained set.

Algorithm 2 SPARK correction loop, Part 2: diagnosis and revisit planning. Neither stage reads the oracle.

STAGE 3: CRITIC

$S \leftarrow$ SCENESUMMARY$(Z^{(1)})$
for each stored object o **do**
  $\pi_o \leftarrow$ PLAUSIBILITY$(o, Z^{(1)}, S) \in [0, 1]$; **flag** if $\pi_o < 0.6$ or unscored
end for
for each waypoint w **do**
  $\rho_w \leftarrow |H_w^{(1)}|$ / ESTIMATEVISIBLE(I_w); **flag** if $\rho_w < 0.4$
    ▷ unassessable, not flagged, if the estimate is zero
end for
for each level-1 cluster k **do**
  flag if $\mathrm{support}(k) < 2$ **and** $\big(|k| \leq 2$ **or** $\bar{d}(k) \geq 40$ m$\big)$
end for
$\mathcal{R}^{(1)} \leftarrow$ union of the three flagged sets

STAGE 4: REVISIT PLANNING

merge flagged clusters within 25 m that share a source waypoint
for each region $r \in \mathcal{R}^{(1)}$ **do**
  $g \leftarrow \|\mathbf{x}_{\text{drone}} - \mathbf{x}_r\|_{xy}$
  if $g > 15$ m **then** emit FORWARD 5 m
  else if $g < 6$ m **then** emit BACK 5 m
  end if
  emit LATERAL-L and LATERAL-R at 5 m
  set altitude to the *nominal* capture height for that waypoint
    ▷ not the degraded height flown in Episode 1
end for
$\mathcal{V}^{(2)} \leftarrow$ emitted candidates, each yawed to face the region centroid

Algorithm 3 SPARK correction loop, Part 3: corrective episode, merge, and development-time evaluation.

STAGE 5: EPISODE 2 AND CORRECTIVE MERGE

fly $\mathcal{V}^{(2)}$ with the Stage-1 motion routine; skip unreachable candidates and record them as skipped
build incoming objects with the Stage-2 pipeline
$T \leftarrow$ types observed in Episode 2, excluding unidentified captions
$A \leftarrow \{\tau : \mathrm{count}(\tau) \geq \max(2, \lceil 0.5 \max_\tau \mathrm{count}(\tau) \rceil)\} \cup \{\tau$ matching an unflagged anchor$\}$
supersede a flagged anchor type only if observed ≥ 2 times in Episode 2
drop a flagged node whose type is unsupported by Episode 2
suppress incoming objects whose type $\notin A$
leave every region outside $\mathcal{R}^{(1)}$ unchanged
$H^{(2)} \leftarrow$ merged graph

STAGE 6: EVALUATION (DEVELOPMENT TIME ONLY)

collect the oracle at *baseline* poses under the same FOV and envelope, with ray-cast occlusion
score critic quality as region-level precision, recall, F1
generate the fixed question set from oracle-supported regions
score the same questions on $H^{(1)}$ and $H^{(2)}$; report the paired difference

B.2 CAPTIONING AND OBJECT REJECTION

Each retained mask m is presented to `gpt-5-mini` in three renderings—a tight crop, the same crop with non-target pixels suppressed, and the full frame with the candidate outlined. These provide local appearance and scene context,

$$(c, v) = \mathrm{VLM}(m_{\text{crop}}, m_{\text{isolated}}, m_{\text{context}}), \tag{4}$$

where c is a label from a closed 22-term vocabulary and v is a validity judgement. A proposal is rejected when the model returns a designated non-object label, the label matches a background term, or a predicted *person* fails a verification call. Background filtering uses whole-word matching

so that compounds such as *roadblock* and *sandbag* are not rejected. This filtering prevents surface fragments proposed by the permissive segmenter from entering memory as generic objects and inflating subsequent counts.

B.3 Depth, backprojection, and sensor noise

Mask depth is taken as the 75th percentile of the in-mask depth values,

$$d_{\text{planar}} = Q_{75}\big(\{\, D_{e,k}(p) : p \in m \,\}\big), \tag{5}$$

instead of a bounding-box center sample. Using the full mask reduces contamination from background pixels, while the percentile above the median favors the object's surface over foreground bleed near the mask boundary. AirSim returns planar depth, which is converted to slant depth along the viewing ray,

$$d_{\text{slant}} = d_{\text{planar}} / \cos(\alpha), \tag{6}$$

before conversion to world coordinates using the vehicle pose and AirSim's NED convention (Shah et al., 2018). Objects beyond the 40 m sensing envelope are discarded before captioning.

Simulator depth is exact, so we apply a sensor model with quadratic axial error $\sigma_z = 0.002z^2$, Gaussian lateral jitter, and dropout that increases at grazing angles, using error forms reported for consumer depth sensors (Khoshelham & Oude Elberink, 2012; Nguyen et al., 2012). Pose drift is not added because pose is the controlled variable in this study.

B.4 Models

Segmentation uses SAM2 (Ravi et al., 2025) through a hosted endpoint. Captioning, validity judgement, the semantic critic, cluster naming and summaries, and the EQA judge all use `gpt-5-mini`. We refer to calls with image inputs as VLM calls and text-only calls as LLM calls. Label comparison for clustering and oracle matching uses `text-embedding-3-small` embeddings.

B.5 Clustering

Detections are pooled across waypoints and clustered by complete-linkage agglomeration using the level-dependent threshold in Algorithm 1. The first level serves as the spatial unit for the critic and evaluation. Embodied-RAG describes clustering using a weighted combination of spatial and semantic similarity. In its released implementation, only positions are passed to the clustering routine, with semantics introduced after clustering through cluster summaries. We restore semantic similarity as a constraint on merging and add a co-visibility constraint.

The semantic constraint blocks merges between nodes describing different object types. Labels are compared by cosine similarity of `text-embedding-3-small` embeddings at a threshold of 0.40. Exact string matching would make descriptions such as *corrugated metal fence panel* and *metal panel perimeter fence* disagree despite referring to the same type of object. Embeddings for all graph labels are computed in a single warmed call. If a similarity is unavailable, it does not block the merge, so a missing embedding cannot silently split a cluster.

The co-visibility constraint blocks a merge when no single waypoint observed both objects. Purely spatial clustering can join nearby objects that were never seen together, including objects on opposite sides of a wall or approached from different directions. We observed this when objects from a flagged waypoint were absorbed into clusters anchored elsewhere, leaving the flagged waypoint without a revisit region. The constraint is symmetric and therefore introduces no processing-order dependence. A pair is blocked only when both nodes carry waypoint provenance, while objects without recorded provenance are clustered by distance alone. Clusters above the first level do not retain waypoint provenance, so the constraint applies at the first level and during the singleton-adoption pass that could otherwise reattach remaining nodes by distance.

Both constraints are implemented by raising the pairwise distance of a blocked pair beyond every linkage threshold, preventing complete-linkage agglomeration from joining them. Each constraint can be disabled independently for ablation. On Hangar, co-visibility reduced cross-waypoint clusters from two to zero without changing the total cluster count.

B.6 Resulting error modes

The resulting memory retains realistic perception errors, including merged adjacent instances, single objects split across masks, coarse labels for small or partially occluded objects, and localization errors of several meters from single-frame backprojection.

C Metric Definitions

C.1 Critic thresholds

Plausibility scores each stored object on $[0, 1]$ according to whether its label is visually supported in context. Entries below 0.6 or without a score are flagged. *Density* compares the number of objects estimated to be visible in a waypoint image with the number stored from that waypoint. The waypoint is flagged when this ratio falls below 0.4 and marked unassessable when the visible-object estimate is zero. *Graph-support coverage* flags a first-level cluster with fewer than two contributing observations when it is also sparse, with at most two member nodes, or far, with a mean source distance of at least 40 m. The distance threshold matches the 40 m sensing envelope, while the additional sparsity or distance condition prevents low support alone from flagging every small cluster in a sparse scene.

C.2 Question construction and scoring

Questions are anchored to a source waypoint and the neighborhood around its cluster. The search radius is the cluster radius plus a 10 m tolerance, with a minimum of 12 m. Answers are generated from graph nodes within that region, with nearby duplicate observations consolidated so repeated views of the same object do not inflate counts.

The question set contains four families that capture different forms of information loss. *Inventory* asks how many objects were visible from the waypoint and how many belonged to the second most common type. Counts use a seven-band scale from *none* through *more than 20*. The finer scale avoids treating memories with, for example, eight and fifty objects as the same *many (10+)* answer under a five-band scale. *Dominant type* asks which object type was most common and whether the runner-up was present, capturing cases where degradation preserves incidental objects but loses the dominant content of the view. *Type richness* asks how many distinct types were visible, using four bands to capture losses in diversity even when some objects remain. *Rare presence* asks whether a type observed exactly once from that waypoint was present. Negative questions use a type absent from the waypoint but frequent elsewhere on the map, with distractors rotated across waypoints to avoid making the answer predictable from object type alone.

A fifth family asked which distance band contained the nearest instance of a type. It was dropped after scoring near zero on both pre- and post-correction memory, indicating that it measured retrieval behavior more than the quality of the stored information.

The same fixed question set is used to score $H^{(1)}$ and $H^{(2)}$. A language-model judge assigns each answer *correct* (1.0), *partial* (0.5), or *wrong* (0.0). For banded questions, an answer one adjacent band from the reference receives partial credit. Unscorable questions are excluded from the mean and reported separately.

C.3 Question set composition

The fixed set contains 151 questions: 32 on Old Shipyard, 40 on Hangar, 39 on Construction Site, and 40 on Downtown West. Each map contributes 16 inventory questions and four type-richness questions. The remaining questions are divided between dominant type and rare presence according to how many types appear exactly once at a waypoint.

Questions are restricted to object types present in both the oracle and the undegraded reference KB for each map. This leaves 5 types on Old Shipyard, 8 on Hangar, 9 on Construction Site, and 6 on Downtown West. The restriction accounts for the different naming schemes used by the two pipelines. The oracle names objects by simulator asset, while the perception pipeline uses a

closed caption vocabulary, so questions expressed only in oracle terms may be unanswerable by any memory the perception pipeline could produce.

Each experimental condition is scored three times and we report the mean. Across all conditions in the paper, the standard deviation across repeats averages 0.0035, indicating that variation from the EQA judge is small relative to the reported changes in memory quality.

C.4 Label agreement

Where explicit label agreement is required, we compare `text-embedding-3-small` embeddings using cosine similarity with a threshold of 0.40. On the calibration set, this accepted 16 of 17 same-object phrasing pairs and none of the 10 different-object pairs. The highest different-object similarity was 0.357, while one same-object pair at 0.373 fell below the threshold.

D Observation and Viewpoint Details

D.1 Simulator and capture configuration

Two AirSim capture profiles are used, and the simulator is restarted when switching between them because capture settings are read at startup. Both render at 1280×720 with a 90° horizontal field of view, corresponding to a 58.7° vertical field of view and a 29.4° vertical half-angle. The perception profile exposes scene RGB and perspective depth, while the oracle profile additionally exposes planar depth and an actor-identity segmentation channel for visibility testing. Matching resolution and field of view across profiles allows direct comparison between their geometry and imagery.

D.2 Motion and capture

For each source or revisit pose, the vehicle climbs to a 20 m transit altitude, translates to the target, descends to capture height, sets yaw on the airframe and pitch on the camera gimbal, stabilizes, and records one observation. Pitch is controlled through the gimbal because the flight controller re-levels the airframe within roughly two seconds, making body-pitch commands unsuitable for a stable capture pose. No observations are recorded in transit, so coverage depends on the evaluated viewpoints and not on the route between them.

D.3 Viewpoint conditions

Baseline leaves each hand-selected pose unchanged. Pitch-up places the bottom of the image 11.3° above the horizon, which with the 29.4° vertical half-angle gives a camera pitch of $+40.7^\circ$ on every map. Defining the degradation by the image boundary instead of a fixed pitch offset keeps the loss of near-field coverage comparable across source poses with different initial pitches. Low-lateral lowers the camera to near ground level and applies a small yaw perturbation, with height adjusted per waypoint where required by the terrain. We do not use a downward-pitch degradation because at these capture heights it largely replaces the scene with nearby pavement that still segments cleanly, while pitch-up removes object-bearing near-field content.

D.4 Transit on Downtown West

Downtown West uses pose-stepping transit because tree geometry makes the standard controller unreliable on that map. Pose stepping affects only movement between capture poses and does not change how observations are recorded, so memory receives observations captured in the same way as on the other maps. All Downtown West episodes use the same transit procedure, including Episode 1, keeping each corrective episode consistent with the memory it corrects.

D.5 Map and waypoint selection

The four maps span distinct levels of clutter and enclosure while remaining small enough for individual correction traces to be inspected. Each map-condition evaluation includes an oracle collection, a perception Episode 1, memory construction, a critic pass, a planned Episode 2, a corrective merge,

and fixed-set evaluation. Within each map, the four source waypoints were selected for object diversity, object density, and regional isolation, so each frames a recognizable local area that provides a meaningful revisit target.

D.6 ORACLE COLLECTION

The oracle uses the same image resolution, field of view, source locations, and 40 m sensing envelope as the perception pipeline, but always observes from the baseline camera pose. Visibility is tested by ray casting against the segmentation channel so that actors hidden behind geometry are not recorded. The resulting reference is not a canonical semantic annotation because simulator asset names can differ from ordinary object names, and scene queries can return decals, lighting props, and other elements not directly represented by the perception vocabulary.