SAGE-VQA: A TRAINING-FREE, SELF-EVOLVING AGENTIC FRAMEWORK WITH GROUNDED EVIDENCE FOR KNOWLEDGE-BASED VISUAL QUESTION ANSWERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge-based visual question answering (KB-VQA) aims to combine visual information with external knowledge to answer user questions. Existing methods face two main bottlenecks: visual perception errors can propagate into hallucinations, and generalization across scenarios remains limited. To address these issues, we propose SAGE-VQA (Self-evolving Agent framework with Grounded Evidence), a training-free agentic framework that combines evidence verification with experience evolution. Our key insight is to explore an Evidence-Driven Adaptive Reasoning (EDAR) strategy. It accurately acquires and verifies question-relevant fine-grained visual information. It also extracts and refines reusable experience to support robust reasoning across tasks. Specifically, we design a Co-verified Visual Grounding module that verifies question-relevant visual information and provides grounded evidence for subsequent reasoning. We further introduce a Self-evolving Hierarchical Memory (SHM) mechanism. SHM organizes experiences into declarative and procedural memories based on their nature. These memories can be reused in subsequent reasoning and continually refined through feedback, without updating model parameters. Experimental results show that our method outperforms competing methods based on lightweight open-source vision-language models (VLMs) as well as those based on closed-source VLMs. Moreover, our method can serve as a plug-and-play reasoning framework for existing VLMs without modifying their architectures.

1 INTRODUCTION

Knowledge-based visual question answering (KB-VQA) aims to combine images and external knowledge bases to answer user questions (Marino et al., 2019; Schwenk et al., 2022; Lu et al., 2022). Images provide visual evidence relevant to the question, and external knowledge bases provide additional information. Closed-source vision-language models (VLMs) achieve strong performance on VQA tasks due to their powerful reasoning capabilities (Yue et al., 2024). However, their reliance on remote APIs increases usage costs and limits personalized adaptation and local deployment. Therefore, improving the knowledge question-answering capabilities of lightweight open-source VLMs that support local deployment has practical value.

Existing KB-VQA methods fall into two broad categories: training-based and training-free methods. Training-based methods adapt to specific KB-VQA tasks by fine-tuning models or training auxiliary modules (Zhang et al., 2024b; Yang et al., 2025b). For example, MR-MKG trains a knowledge encoder and cross-modal alignment modules to improve multimodal knowledge reasoning. (Lee et al., 2024). These methods are effective on specific datasets but require extensive manual annotation and substantial training costs. Task-specific fine-tuning also limits their generalization to other tasks and domains. This motivates research into improving the reasoning abilities of lightweight open-source VLMs without task-specific fine-tuning.

Training-free methods improve VQA through prompt enhancement and tool use. For example, PICa combines image captions, tags, and example question–answer pairs into prompts. It uses the

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.

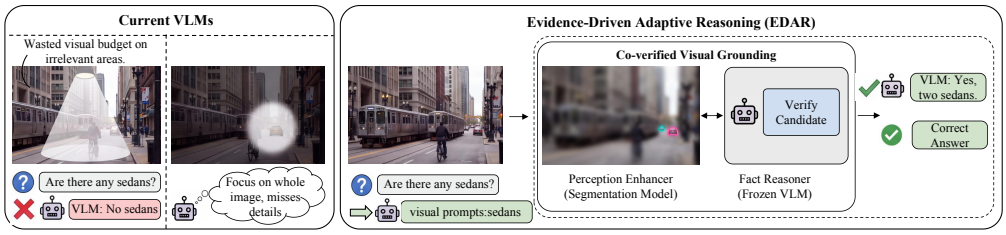


Figure 1: Motivation for EDAR. Whole-image processing can allocate visual capacity to question-irrelevant regions and miss answer-relevant local details. Co-verified Visual Grounding locates and verifies candidate regions before answer generation.

knowledge of VLMs to answer questions (Yang et al., 2022). AVIS dynamically calls tools during reasoning to obtain the information needed for answering (Hu et al., 2023b). However, reasoning in these methods still depends on the accuracy of visual information. Existing VLMs can still produce object hallucinations and fine-grained visual perception errors (Hoang et al., 2026; Li et al., 2026). As shown in Figure 1, a VLM processes the entire image. Part of its visual processing budget is spent on question-irrelevant objects and backgrounds (Han et al., 2023; Brauwers & Frasincar, 2023). This limits its attention to key objects. Whole-image representations may also fail to focus sufficiently on local details relevant to the answer. These two limitations can cause visual information to be missed or misinterpreted and affect subsequent reasoning. This raises the first challenge: how can we acquire question-relevant visual evidence without updating model parameters? The goal is to reduce the impact of perception errors on subsequent reasoning.

In visual question answering, questions of the same type may share similar reasoning experience. It is therefore important to retain useful knowledge and solution methods for later reasoning on similar questions. Existing studies store reflections, past experience, or executable skills to support experience reuse in model reasoning (Shinn et al., 2023; Park et al., 2023; Wang et al., 2023; Xu et al., 2025). For example, ReasoningBank distills reusable reasoning strategies from successful and failed task trajectories. It retrieves relevant experience for later tasks and appends newly extracted memories to its memory bank (Ouyang et al., 2026). However, past experience may not apply to new, similar questions. Simply adding more memories therefore creates redundancy and makes accurate retrieval difficult. It also reduces the applicability of existing experience. This raises the second challenge: how can a model reuse relevant knowledge and methods for new questions and continuously refine past experience through feedback? How can this process support accurate memory retrieval and robust use across different question types?

To address these two challenges, we propose SAGE-VQA (Self-evolving Agent framework with Grounded Evidence). This training-free agentic framework combines evidence verification with experience evolution. Our key insight is to explore an Evidence-Driven Adaptive Reasoning (EDAR) strategy. The strategy aims to accurately acquire question-relevant fine-grained visual information and use refined experience to support robust reasoning across tasks. To address the first challenge, we design a Co-verified Visual Grounding module to verify visual information. A pretrained segmentation model serves as the perception enhancer and provides question-relevant visual candidates from the image. A frozen VLM serves as the fact reasoner and checks semantic consistency between the target objects and these visual candidates. To address the second challenge, we introduce a Self-evolving Hierarchical Memory (SHM) mechanism. SHM organizes experience by its nature and use into declarative and procedural memory. Declarative memory includes episodic records of reasoning and corrections, as well as semantic rules drawn from similar episodes. Procedural memory stores executable methods and their conditions of use. To initialize the bank, the system extracts reusable experience from selected training cases. For later questions, the system retrieves relevant memories, and SHM refines the bank using post-answer feedback. Finally, EDAR combines visual evidence, knowledge from a multimodal knowledge graph, and relevant memories for reasoning. It assesses readiness at each iteration to determine whether the available evidence is sufficient to answer the user's question.

Our contributions are as follows:

- We propose SAGE-VQA, a training-free framework that uses EDAR to combine question-relevant visual verification with self-evolving experience. It improves the KB-VQA performance of lightweight open-source VLMs without task-specific parameter training.
- We design Co-verified Visual Grounding, which uses a pretrained segmentation model as a perception enhancer to obtain question-relevant visual candidates and a frozen VLM to assess their semantic consistency with target objects. This strengthens perception of relevant visual details before downstream reasoning.
- We introduce SHM to support the reuse and feedback-based refinement of declarative and procedural memories, allowing the framework to improve past experience without updating model parameters.
- We validated SAGE-VQA on four datasets. The results demonstrate that our approach achieves state-of-the-art performance, outperforming both lightweight open-source VLMs of comparable scale and proprietary closed-source models.

2 Related Work

Training-based KB-VQA. Training-based methods optimise backbone or auxiliary parameters to associate visual evidence with task-relevant knowledge. Early approaches such as KAT combine implicit language-model knowledge with retrieved explicit knowledge, providing complementary sources for KB-VQA (Gui et al., 2022). MuKEA organises visual objects, implicit relations, and factual answers into multimodal triplets, accumulating retrievable knowledge for visual–fact reasoning (Ding et al., 2022). REVIVE incorporates object regions and their relations into knowledge retrieval and answer prediction, strengthening the use of local visual evidence in KB-VQA (Lin et al., 2022). Subsequent work broadens knowledge integration: REVEAL jointly learns memory encoding, retrieval, and generation to incorporate image–text, question–answer, and knowledge-graph information into KB-VQA (Hu et al., 2023c). Later VLM-based methods train auxiliary components around frozen model backbones. MR-MKG trains multimodal knowledge encoders and cross-modal alignment modules to make visual and semantic graph relations accessible to language models (Lee et al., 2024). QACap trains a question-aware captioner that translates answer-relevant visual details into text for a frozen language model (Yang et al., 2025b). These developments improve knowledge access and visual information transfer, but freezing backbones does not remove the need for additional parameter optimisation and suitable training data.

Training-free KB-VQA. Training-free methods combine frozen pretrained models through prompting, retrieval, and tool use, without additional parameter training or adaptation of any component. Early approaches such as PICa use captions, tags, and example question–answer pairs to invoke language-model knowledge for KB-VQA (Yang et al., 2022). PnP-VQA exposes question-relevant visual details through local captions guided by the input question (Tiong et al., 2022). Img2LLM constructs captions and example question–answer pairs from images, providing both visual context and task demonstrations (Guo et al., 2023). Their effectiveness depends on preserving answer-relevant visual details in the textual context. Subsequent approaches integrate external information acquisition with reasoning to supplement parametric knowledge. AVIS dynamically selects visual tools and retrieval operations based on intermediate results, allowing KB-VQA systems to actively seek missing information (Hu et al., 2023b). MG^2-RAG organises textual entities, visual regions, and document chunks into a multi-granularity graph, retrieving structured evidence for visual questions requiring multiple knowledge connections (Dai et al., 2026). However, retrieval relevance alone does not establish evidential support for an answer. Retrieved information inconsistent with the image, or conflicting visual observations, can still interfere with reasoning. Assessing the validity and consistency of visual evidence alongside knowledge acquisition therefore remains an important challenge in KB-VQA. Beyond individual queries, early memory-augmented agents retain feedback or skills for subsequent reuse. Reflexion uses reflections on previous attempts to inform later decisions, while Voyager reuses acquired capabilities through a library of executable skills (Shinn et al., 2023; Wang et al., 2023). Later, A-MEM dynamically links and revises historical memories; Mem^p retrieves trajectories and abstract procedural steps as reasoning context and refines memories through feedback (Xu et al., 2025; Fang et al., 2026). These general-purpose agents offer mechanisms that can inform experience accumulation and reuse in KB-VQA. However, semantic similarity to a previous problem does not guarantee that its knowledge or procedures apply to a new image. Aligning experience reuse with current visual evidence therefore remains an open question.

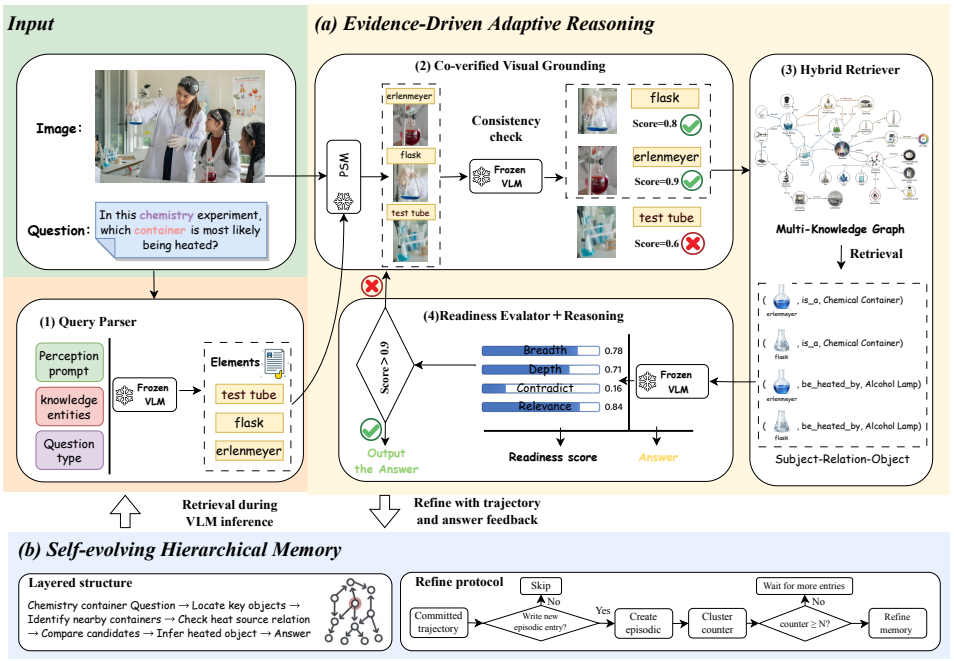


Figure 2: Overview of SAGE-VQA. The framework comprises two core modules: Co-verified Visual Grounding for verifying question-relevant visual targets, and SHM for feedback-driven memory construction and refinement. EDAR integrates visual evidence, external knowledge, and relevant memories, then uses readiness assessment to guide further evidence gathering or answer generation.

3 METHOD

3.1 FRAMEWORK OVERVIEW

SAGE-VQA is a training-free agentic framework that combines evidence verification with experience evolution. Figure 2 illustrates the framework. It uses two modules: Co-verified Visual Grounding and Self-evolving Hierarchical Memory (SHM). Co-verified Visual Grounding locates and verifies question-relevant visual targets. SHM builds and refines a memory bank from post-answer feedback for use on later questions. The Evidence-Driven Adaptive Reasoning (EDAR) strategy acquires and verifies visual evidence, reasons with external knowledge and relevant memories, and uses readiness assessment to decide whether to seek further evidence or answer. Task notation and core prompt templates appear in Appendices A.1 and A.2, respectively.

3.2 CO-VERIFIED VISUAL GROUNDING

Insight. A model may fail to identify question-relevant objects, attributes, or relations in an image. Subsequent reasoning can then rely on incorrect visual information and produce an unreliable answer. These errors can also mislead external knowledge retrieval. We therefore design Co-verified Visual Grounding to locate and verify local candidates before using visual information for reasoning.

Open-vocabulary visual prompts. A frozen VLM serves as the fact reasoner $\mathcal{M}$. At iteration t, $\mathcal{M}$ provides a set of open-vocabulary visual prompts $\mathcal{P}_t$. Each prompt describes a target object in text and guides localization in the image. Below, task subscripts on f distinguish calls to the same frozen model $\mathcal{M}$ for different tasks.

Co-verified Visual Grounding. A pretrained segmentation model serves as the perception enhancer $\mathcal{S}_{\text{seg}}$. It uses open-vocabulary visual prompts to locate and segment candidate regions in image I. The reasoner $\mathcal{M}$ checks whether these regions match the concepts in the prompts to assess semantic consistency. Let $p_{t,j}$ denote a prompt used at iteration t. Let $m_{t,j,k}$ and $b_{t,j,k}$ denote the mask and bounding box of its k-th candidate. Candidate verification is written as:

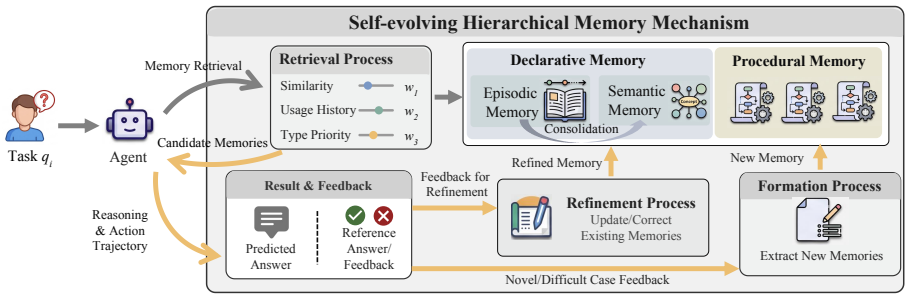


Figure 3: Self-evolving Hierarchical Memory (SHM). SHM organizes episodic records and semantic rules as declarative memory, and executable methods as procedural memory. Based on feedback, SHM can refine existing memories or add new memory.

$$(a_{t,j,k}, v_{t,j,k}) = f_{\mathrm{v}}(I[m_{t,j,k}], p_{t,j})\,. \tag{1}$$

where, f_{v} denotes visual candidate verification. The index $t \in \{1, \ldots, T\}$ denotes the iteration within one question, and T is the maximum number of iterations. The subscript t identifies the prompts, candidate regions, and verification results used at iteration t. Unchanged information is reused without repeating segmentation or verification. The index j identifies a prompt, and k identifies a candidate for that prompt. The term $I[m_{t,j,k}]$ denotes the masked crop. The value $a_{t,j,k} \in \{0, 1\}$ records the semantic match, and $v_{t,j,k} \in [0, 1]$ is the model-reported confidence.

The system retains the candidates and their verification annotations as $\mathcal{A}_{t,j} = \{(m_{t,j,k}, b_{t,j,k}, a_{t,j,k}, v_{t,j,k})\}_{k=1}^{K_{t,j}}$, where $K_{t,j}$ is the number of candidates. The reasoner $\mathcal{M}$ then compares the candidates with its whole-image observations. It checks for disagreements in objects, counts, and relations. Candidate records and whole-image observations jointly form the visual evidence $\mathcal{V}_t$. Failed candidates retain their failure annotations so the reasoner can assess their impact. Appendix A.3 details candidate verification, whole-image comparison, and visual rechecking.

3.3 Self-evolving Hierarchical Memory Mechanism

Memory organization and retrieval. Inspired by the episodic–semantic distinction (Tulving, 1972) and memory organization in language agents (Sumers et al., 2024), SHM organizes its bank into declarative and procedural memory (Figure 3). Declarative memory combines episodic memory of task-specific reasoning and corrections with generalizable semantic memory distilled from related episodes. Procedural memory stores executable solution methods and their applicability conditions. The initial bank is built from training examples disjoint from evaluation using reasoning trajectories and post-answer feedback from reference answers, with model parameters fixed. For each question, SHM retrieves relevant entries from the latest completed bank based on similarity to memory titles, usage history, and memory type. Appendices A.4 and A.5 detail the memory representations, cold-start construction, and retrieval procedure.

Memory refinement. After answering a question, the system receives feedback based on the reference answer y_i. Let τ_i be the actual number of iterations and Π_i the complete reasoning record. The current record contains Q_i, Π_i, the prediction $\hat{A}_i$, the reference answer, and correctness $c_i \in \{0, 1\}$ under the task protocol. A correct answer reinforces the usage statistics of relevant memories. An incorrect answer triggers refinement of entries that remain applicable but need revision:

$$\mu'_\ell = f_{\mathrm{m}}\Big(\mu_\ell, Q_i, \Pi_i, \hat{A}_i, y_i\Big)\,. \tag{2}$$

where, f_{m} refines an existing memory using the current case and feedback. The term μ'_ℓ denotes the refined entry. Refinement can correct knowledge descriptions, add applicability conditions, or revise program content. An inapplicable retrieved entry calls for an adjustment to its retrieval priority

rather than a content rewrite based on that case. The original entry is retained if refinement produces no valid revision. This mechanism improves experience for similar questions without adding every reasoning record as a new memory. It therefore limits redundant growth of the bank.

New memory formation and hierarchical organization. Let $\mathcal{U}_i$ denote the memories selected for the current question. New experience can be extracted when this set is empty and the answer is incorrect or correct after multiple iterations:

$$\eta_i = \mathbf{1}\left[\mathcal{U}_i = \varnothing \ \wedge \ \left(c_i = 0 \ \vee \ (c_i = 1 \wedge \tau_i > 1)\right)\right]. \tag{3}$$

where $\mathbf{1}[\cdot]$ is the indicator function. The condition $\eta_i = 1$ permits extraction; only valid content is inserted. Episodes sharing a capability gap and similar solution patterns can form semantic memory once the group meets the size requirement. Recurring computational needs can support the formation or refinement of procedural memory. The bank update is:

$$\mathcal{B}^{(s+1)} = h\Big(\{\mu \in \mathcal{B}^{(s)} : \mu \notin \mathcal{F}_i\} \cup \mathcal{F}_i' \cup \mathcal{N}_i\Big). \tag{4}$$

where h organizes the memory set into the hierarchy under the conditions above. $\mathcal{B}^{(s)}$ denotes the current write version, $\mathcal{F}_i$ the entries to replace, $\mathcal{F}_i'$ their valid revisions, and $\mathcal{N}_i$ valid new entries. Algorithm 1 details this update.

3.4 Evidence-Driven Adaptive Reasoning Strategy

Building on visual verification and hierarchical memory, EDAR brings visual evidence, external knowledge, and relevant memories into one reasoning process. It decides whether to seek more evidence or answer based on evidence sufficiency. EDAR handles adaptive reasoning within a query. After an answer is recorded, SAGE-VQA passes the completed case to SHM when feedback updates are enabled and a reference answer is available. The reference answer does not inform the current prediction. Algorithm 2 in Appendix A.6 details the reasoning and feedback handoff.

Initialization and multimodal evidence acquisition. Given image I and question Q, $\mathcal{M}$ obtains an open-vocabulary visual prompt set $\mathcal{P}_1$ and a knowledge entity set $\mathcal{E}_1$ from the question and global image description C. We write $(\mathcal{P}_1, \mathcal{E}_1) = f_{\mathrm{p}}(Q, C)$, where f_{p} denotes question parsing. The prompts are passed to Co-verified Visual Grounding in Section 3.2 to form visual evidence $\mathcal{V}_t$. The entities are used to query the multimodal knowledge graph. The system retrieves relevant memories $\mathcal{U}_Q$ using Section 3.3. Procedural memory executes when its input requirements are met. Inputs may come from the question, image descriptions, or acquired visual information. The set $\mathcal{W}_t$ contains executed program outputs that pass output checks.

The system queries the multimodal knowledge graph $\mathcal{G}$ using knowledge entities. It also uses locally verified image regions for visual retrieval. Let $\mathcal{E}_t$ and $\mathcal{O}_t$ denote the entities and image regions used for retrieval at iteration t. The knowledge update is:

$$\mathcal{K}_t = \mathcal{K}_{t-1} \cup g(\mathcal{E}_t, \mathcal{O}_t). \tag{5}$$

where, g denotes retrieval from the $\mathcal{G}$ using entities and image regions. The set $\mathcal{K}_t$ contains the graph knowledge acquired through iteration t. Entity retrieval returns relevant relations and facts. Visual retrieval locates graph entities through similar images. The initial retrieval uses both channels. Later queries target missing information. Appendix A.7 describes both retrieval channels.

Readiness assessment and answering. $\mathcal{M}$ assesses whether the visual evidence, graph knowledge, relevant memories, validated program outputs, and reasoning record are sufficient to answer the question. It assigns a readiness score between 0 and 1 and identifies unresolved issues in breadth, depth, contradiction, or relevance. If readiness remains below the threshold and iterations remain, $\mathcal{M}$ uses these issues and the reasoning record to select visual rechecking or knowledge supplementation. Visual rechecking can reuse existing segmentation results. The system reassesses readiness after each iteration and answers using this information once readiness reaches the threshold or the iteration budget is exhausted. Appendix A.8 details the issue categories and stopping conditions.

Table 1: ScienceQA test-set comparison. TF: training-free across all components. Subject: NAT/SOC/LAN denote natural/social/language science; context modality: TXT/IMG/NO denote text/image/no context; grade: G1–6/G7–12; Avg.: average accuracy.

Method	TF	Subject			Context Modality			Grade		Avg.
		NAT	SOC	LAN	TXT	IMG	NO	G1-6	G7-12	
Human (Lu et al., 2022)	—	90.23	84.97	87.48	89.60	87.50	88.10	91.59	82.42	88.40
Multimodal-CoT (Zhang et al., 2024b)	✗	91.03	93.70	86.64	90.13	88.25	89.48	91.12	89.26	90.45
LLaVA (Liu et al., 2023)	✗	90.36	95.95	88.00	89.49	88.00	90.66	90.93	90.90	90.92
LLaVA+GPT-4 judge (Liu et al., 2023)	✗	91.56	96.74	91.09	90.62	88.99	93.52	92.73	92.16	92.53
CalibRL (Qwen2.5-VL-7B) (Huang et al., 2026)	✗	-	-	-	-	-	-	-	-	89.04
LLaMA-Adapter (Zhang et al., 2024a)	✗	84.37	88.30	84.36	83.72	80.32	86.90	85.83	84.05	85.19
LaVIN-13B (Luo et al., 2023)	✗	89.88	94.49	89.82	88.95	87.61	91.85	91.45	89.72	90.83
MR-MKG (FLAN-UL2-19B) (Lee et al., 2024)	✗	95.74	90.33	92.00	95.50	92.41	93.31	93.98	93.01	93.63
RG-VQA (Bunny-8B) (Sravanthi et al., 2025)	✗	90.45	96.85	91.64	95.37	88.50	93.24	92.47	91.43	92.10
MG^2-RAG(Qwen3.5-27B) (Dai et al., 2026)	✓	98.45	95.28	98.73	98.44	96.43	98.68	98.42	96.84	97.85
Gemini-3.1-Pro (Dai et al., 2026)	✓	93.53	94.12	90.81	92.98	94.25	90.27	94.03	90.93	92.90
GPT-5.2 (Dai et al., 2026)	✓	97.60	93.05	95.96	97.73	95.35	96.17	96.54	95.88	96.30
SAGE-VQA (GPT-4o-mini)	✓	88.90	85.94	90.00	86.95	83.94	91.64	90.86	84.44	88.56
SAGE-VQA (Qwen3.5-9B)	✓	95.29	91.79	98.27	94.23	91.57	98.40	96.84	92.62	95.33
SAGE-VQA (Qwen3.5-27B)	✓	98.67	96.85	**99.73**	**98.68**	97.47	**99.51**	98.97	97.82	98.56
SAGE-VQA (Qwen3.6-35B)	✓	**99.02**	**97.86**	99.55	**98.68**	**98.26**	98.75	**99.27**	**98.29**	**98.92**

4 EXPERIMENTS

4.1 SETUP

We evaluate four settings: ScienceQA (Lu et al., 2022) on its official test split; all 1,145 A-OKVQA (Schwenk et al., 2022) validation questions in both answer formats; all 900 MMMU (Yue et al., 2024) validation questions; and 1,000 validation questions from MicroQuant-VQA, a private dataset we constructed(appendix B). We use top-1 accuracy to evaluate multiple-choice questions and VQAv2 soft accuracy to evaluate direct-answer tasks. Appendix C provides full statistics.

We use Qwen3.5-9B as the main backbone and SAM3 (Carion et al., 2026) as the perception enhancer. We also evaluate Qwen3.5-27B, Qwen3.6-35B, and GPT-4o-mini. The memory bank is initialized from data disjoint from the evaluation set. It continues to evolve during evaluation. Later questions use the latest completed memory version. Model parameters remain fixed. Further implementation details are provided in Appendices D and E.

4.2 COMPARISON RESULTS

Overall results. Tables 1–3 compare SAGE-VQA with training-based and training-free methods. With a lightweight Qwen3.5-9B backbone, SAGE-VQA outperformed MR-MKG on ScienceQA by 1.70 percentage points. MR-MKG uses a frozen FLAN-UL2-19B backbone and requires task-specific training of auxiliary modules. On A-OKVQA multiple choice, our 9B configuration exceeded V3Fusion by 0.81 percentage points. V3Fusion trains an MLP to fuse a three-model ensemble that includes 13B and 7B LLaVA backbones. On MMMU, our 9B configuration surpassed MAGIC-VQA with GPT-4o-mini by 5.27 percentage points. MAGIC-VQA additionally trains a lightweight GNN for commonsense reasoning. The same 9B configuration also exceeded Gemini-3.1-Pro on ScienceQA and QACap+Claude-3.5 on A-OKVQA multiple choice by 2.43 and 13.28 percentage points, respectively. On A-OKVQA direct answers, it surpassed PromptCap+GPT-3 by 12.61 percentage points. PromptCap combines a trained captioner with GPT-3. The 27B configuration further outperformed GPT-5.2 on ScienceQA by 2.26 percentage points. These comparisons show that SAGE-VQA can outperform methods requiring auxiliary-component training or closed-source models while using a lightweight open-source reasoning backbone. The direct-answer gains extend this advantage beyond selecting from predefined options.

MicroQuant-VQA. We further evaluated SAGE-VQA on MicroQuant-VQA, which requires quantitative measurements from micro-CT images. As shown in Table 13 (Appendix F.1), the 9B and 27B configurations achieved 35.80% and 87.50%, improving over their bare backbones by 12.70

Table 2: A-OKVQA validation. MC/DA: multiple choice/direct answer. QACap's occurrence-based DA score is omitted as non-comparable.

Method	TF	MC	DA
ViLBERT (Schwenk et al., 2022)	✗	49.1	30.6
GPV-2 (Schwenk et al., 2022)	✗	60.3	48.6
MAGIC-VQA (Yang et al., 2025a)	✗	76.55	—
QACap+Claude-3.5 (Yang et al., 2025b)	✗	76.7	—
V3Fusion (Tekin et al., 2026)	✗	89.17	—
PromptCap+GPT-3 (Hu et al., 2023a)	✗	73.2	56.3
SAGE-VQA (GPT-4o-mini)	✓	80.26	61.46
SAGE-VQA (Qwen3.5-9B)	✓	89.98	68.91
SAGE-VQA (Qwen3.5-27B)	✓	90.82	69.84
SAGE-VQA (Qwen3.6-35B)	✓	**91.54**	**72.67**

Table 3: MMMU validation under the official mixed-format protocol. TF: training-free across all components.

Method	TF	Acc.
GPT-4V (Yue et al., 2024)	✓	56.80
GPT-4o-mini (OpenAI, 2024)	✓	59.40
Qwen2.5-VL-7B (Bai et al., 2025)	✓	58.60
MAGIC-VQA (Yang et al., 2025a)	✗	61.03
V3Fusion (Tekin et al., 2026)	✗	56.09
CalibRL (Huang et al., 2026)	✗	56.55
SAGE-VQA (GPT-4o-mini)	✓	64.59
SAGE-VQA (Qwen3.5-9B)	✓	66.30
SAGE-VQA (Qwen3.5-27B)	✓	75.32
SAGE-VQA (Qwen3.6-35B)	✓	**76.55**

Table 4: Incremental ablation on Qwen3.5-9B. The EDAR row adds visual verification and adaptive reasoning to MMKG retrieval. Parentheses report gains over the preceding row.

Configuration	ScienceQA	MMMU
Bare VLM	70.79	50.12
+ MMKG	76.96 (+6.17)	54.44 (+4.32)
+ EDAR	92.76 (+15.80)	61.29 (+6.85)
+ SHM	**95.33** (+2.57)	**66.30** (+5.01)

Table 5: Feedback-driven memory self-evolution on 84 recurring ScienceQA hard cases. Δ is measured from Pass 1; transitions are relative to the preceding pass.

Pass	Correct	Acc. (%)	Δ (pp)	Corrected	Regressed
1	32	38.10	–	–	–
2	38	45.24	+7.14	15	9
3	40	47.62	+9.52	14	12
4	41	**48.81**	**+10.71**	11	10
5	40	47.62	+9.52	9	10

and 58.20 percentage points. These results support the framework's applicability to materials analysis through visual grounding and reusable computational procedures. Appendix B provides the task details.

4.3 Ablation Study

Table 4 presents the ablation results using Qwen3.5-9B as the backbone. MMKG retrieval alone improved performance on both datasets. The EDAR row adds visual verification and adaptive reasoning to that retrieval baseline, thereby completing the EDAR strategy. This stage produced the largest incremental gain in the reported ordering. Adding SHM further improved performance through memory reuse and updating. The 27B ablation in Appendix F.2 showed the same pattern with a stronger backbone.

4.4 Feedback-Driven Memory Evolution

Protocol. Using Qwen3.6-35B, we make five passes over 84 ScienceQA errors from an earlier full-system run, each inheriting the preceding bank. Feedback updates usage statistics and asynchronously refines applicable retrieved entries after errors. Cases without relevant retrieved memory can supply new experience after an error or a correct answer reached through multiple iterations. Only later queries can benefit; model parameters remain fixed.

Result. As shown in Table 5, accuracy increased from 38.10% to a peak of 48.81% in Pass 4. It then declined to 47.62% in the final pass. Across passes, some previously incorrect cases were corrected, while some previously correct cases became incorrect. These results show improved overall accuracy alongside performance regressions, highlighting the need to assess and refine memories during continued use.

4.5 Backbone Scaling Study

Figure 4 evaluates SAGE-VQA across backbone scales and model families under common retrieval and iteration budgets. Within Qwen3.5, scaling from 9B to 27B consistently improves all benchmarks and answer formats, with the clearest benefit on MMMU. Qwen3.6-35B improves further. Results with GPT-4o-mini show that the framework also transfers beyond the Qwen family. Overall,

SAGE-VQA remains effective with lightweight backbones and benefits from stronger ones, indicating that backbone capability and the proposed reasoning framework are complementary.

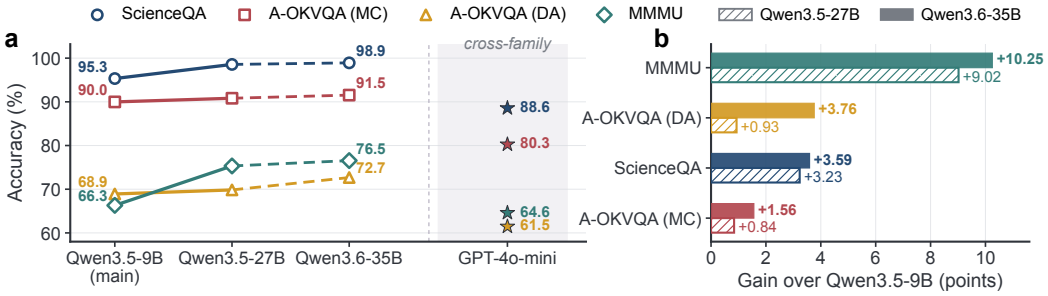
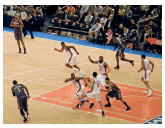
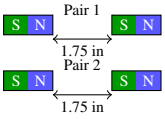


Figure 4: SAGE-VQA performance across fact-reasoner backbones. (a) Accuracy on ScienceQA, A-OKVQA, and MMMU. (b) Absolute gains of the larger Qwen backbones over Qwen3.5-9B.

4.6 Case Studies

Analysis. The cases show that SAGE-VQA routes distinct capability gaps to complementary repairs: entity retrieval, verified grounding, or Procedural Memory. Recovery therefore depends on gap diagnosis and evidence validation. Across cases, correction follows diagnosis, targeted evidence acquisition, and validation before commitment.

Table 6: Representative error recovery.

Question (gold)		Baseline/prior state (wrong)	SAGE-VQA (correct)	Effective component
	Which team is in white/orange? (*Knicks*)	The bare VLM fixates on the opposing dark jersey ("JACKSON," No. 1), never establishes the white/orange team's identity, and predicts Raptors.	Although the VLM misreads a jersey as "UTAH," MMKG retrieval links the image to the New York Knicks. The fact reasoner checks this identity against the choices and orange-white color scheme, yielding Knicks.	**MMKG retrieval** + evidence arbitration.
	How many people have their arms around the woman? (*one*)	It conflates the three riders with the relation-specific count and predicts three, even though its own trace observes that only the man behind the woman has an arm around her.	The VLM separates group membership from the queried relation: the rear man embraces the woman, whereas the front rider does not. SAM3 independently returns one relevant detection, supporting one.	**EDAR** enforces relation-aware counting.
	Which pair has the greater magnetic-force magnitude? (*the same*)	An earlier Qwen3.6-35B pass overweights a local cue and predicts Pair 1, despite identical materials, orientations, and separations.	Procedural memory invokes `compute_magnetic_force_magnitude`. A successful execution compares the verified attributes and equal 1.75-in gaps, yielding the same.	**Procedural-memory execution** + verified measurement.

5 Conclusion

We propose SAGE-VQA, a training-free agentic framework for KB-VQA that combines evidence verification with experience evolution. Co-verified Visual Grounding verifies question-relevant visual information, while SHM organizes declarative and procedural memories and refines them through feedback. The EDAR strategy acquires and verifies visual evidence, retrieves external knowledge and relevant memories, and uses readiness assessment to guide further evidence acquisition or answering. Experiments across diverse benchmarks support the effectiveness of combining evidence verification with experience reuse without updating model parameters.

AI USE STATEMENT

Generative AI tools were used to assist with English-language editing.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Gianni Brauwers and Flavius Frasincar. A general survey on attention mechanisms in deep learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3279–3298, 2023. doi: 10.1109/TKDE.2021.3126456.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, RISHI HAZRA, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollar, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment anything with concepts. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 138846–138923, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/e0982cbc81401df3430ee1ff780dc7a2-Paper-Conference.pdf.

Sijun Dai, Qiang Huang, Xiaoxing You, and Jun Yu. MG2-RAG: Multi-granularity graph for multimodal retrieval-augmented generation. *arXiv preprint arXiv:2604.04969*, 2026.

Yang Ding, Jing Yu, Bang Liu, Yue Hu, Mingxin Cui, and Qi Wu. MuKEA: Multimodal knowledge extraction and accumulation for knowledge-based visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5089–5098, 2022.

Runnan Fang, Yuan Liang, Xiaobin Wang, Jialong Wu, Shuofei Qiao, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. Mem^p: Exploring agent procedural memory. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 17490–17502. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.866. URL https://aclanthology.org/2026.findings-acl.866/.

Google DeepMind Team. Gemini 3.1 pro: Best for complex tasks and bringing creative concepts to life. https://deepmind.google/models/gemini/pro/, 2026.

Liangke Gui, Borui Wang, Qiuyuan Huang, Alexander Hauptmann, Yonatan Bisk, and Jianfeng Gao. KAT: A knowledge augmented transformer for vision-and-language. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 956–968, 2022.

Jiaxian Guo, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Boyang Li, Dacheng Tao, and Steven C.H. Hoi. From images to textual prompts: Zero-shot visual question answering with frozen large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10867–10877, 2023.

Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, Zhaohui Yang, Yiman Zhang, and Dacheng Tao. A survey on vision transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):87–110, 2023. doi: 10.1109/TPAMI.2022.3152247.

Nhat Hoang, Minh Vu, My T. Thai, and Manish Bhattarai. PAS: Prelim attention score for detecting object hallucinations in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18273–18283, 2026.

Yushi Hu, Hang Hua, Zhengyuan Yang, Weijia Shi, Noah A. Smith, and Jiebo Luo. PromptCap: Prompt-guided image captioning for VQA with GPT-3. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2951–2963. IEEE Computer Society, 2023a. doi: 10.1109/ICCV51070.2023.00277.

Ziniu Hu, Ahmet Iscen, Chen Sun, Kai-Wei Chang, Yizhou Sun, David A. Ross, Cordelia Schmid, and Alireza Fathi. AVIS: Autonomous visual information seeking with large language model agent. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023b.

Ziniu Hu, Ahmet Iscen, Chen Sun, Zirui Wang, Kai-Wei Chang, Yizhou Sun, Cordelia Schmid, David A. Ross, and Alireza Fathi. REVEAL: Retrieval-augmented visual-language pre-training with multi-source multimodal knowledge memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23369–23379, 2023c.

Zhuoxu Huang, Mengxi Jia, Hao Sun, Xuelong Li, and Jungong Han. Controllable exploration in hybrid-policy RLVR for multi-modal reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=5wxyCidRsK.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

Amita Kamath, Christopher Clark, Tanmay Gupta, Eric Kolve, Derek Hoiem, and Aniruddha Kembhavi. Webly supervised concept expansion for general purpose vision models. In *Computer Vision – ECCV 2022*, pp. 662–681. Springer Nature Switzerland, 2022. ISBN 978-3-031-20059-5.

Junlin Lee, Yequan Wang, Jing Li, and Min Zhang. Multimodal reasoning with multimodal knowledge graph. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 10767–10782, 2024.

Rang Li, Lei Li, Shuhuai Ren, Hao Tian, Shuhao Gu, Shicheng Li, Zihao Yue, Yudong Wang, Wenhan Ma, Zhe Yang, Jingyuan Ma, Zhifang Sui, and Fuli Luo. GroundingME: Exposing the visual grounding gap in MLLMs through multi-dimensional evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2412–2422, 2026.

Yuanze Lin, Yujia Xie, Dongdong Chen, Yichong Xu, Chenguang Zhu, and Lu Yuan. REVIVE: Regional visual representation matters in knowledge-based visual question answering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Ye Liu, Hui Li, Alberto Garcia-Duran, Mathias Niepert, Daniel Onoro-Rubio, and David S. Rosenblum. MMKG: Multi-modal knowledge graphs. In *European Semantic Web Conference (ESWC)*, 2019.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Gen Luo, Yiyi Zhou, Tianhe Ren, Shengxin Chen, Xiaoshuai Sun, and Rongrong Ji. Cheap and quick: Efficient vision-language instruction tuning for large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Jinghao Luo, Yuchen Tian, Chuxue Cao, Ziyang Luo, Hongzhan Lin, Kaixin Li, Chuyi Kong, Ruichao Yang, and Jing Ma. From storage to experience: A survey on the evolution of LLM agent memory mechanisms. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 41622–41652, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.2069. URL https://aclanthology.org/2026.findings-acl.2069/.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. OK-VQA: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3195–3204, 2019.

Brian H. Miao, Robert J. Headrick, Zhao Li, Leonardo Spanu, David J. Loftus, and Michael D. Lepech. Development of biopolymer composites using lignin: A sustainable technology for fostering a green transition in the construction sector. *Cleaner Materials*, 14:100279, 2024.

OpenAI. GPT-4o mini: Advancing cost-efficient intelligence. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/, 2024.

OpenAI Team. Introducing GPT-5.2: The most advanced frontier model for professional work and long-running agents. https://openai.com/index/introducing-gpt-5-2/, 2026.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T. Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. ReasoningBank: Scaling agent self-evolving with reasoning memory. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/pdf/24a0425aa87cfcd0ba7344bbf5e11cdfe9e497f4.pdf.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST)*, 2023.

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 *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-OKVQA: A benchmark for visual question answering using world knowledge. In *European Conference on Computer Vision (ECCV)*, 2022.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Settaluri Lakshmi Sravanthi, Pulkit Agarwal, Debjyoti Mondal, et al. RG-VQA: Leveraging retriever-generator pipelines for knowledge intensive visual question answering. In *Findings of the Association for Computational Linguistics: EMNLP*, pp. 24048–24060, 2025.

Theodore R. Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L. Griffiths. Cognitive architectures for language agents. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=1i6ZCvflQJ.

Selim Furkan Tekin et al. Vision verification enhanced fusion of VLMs for efficient visual reasoning. *arXiv preprint*, 2026.

Anthony Meng Huat Tiong, Junnan Li, Boyang Li, Silvio Savarese, and Steven C.H. Hoi. Plug-and-play VQA: Zero-shot VQA by conjoining large pretrained models with zero training. In *Findings of the Association for Computational Linguistics: EMNLP*, pp. 951–967, 2022.

Salvatore Torquato. *Random Heterogeneous Materials: Microstructure and Macroscopic Properties*. Springer Science & Business Media, 2002.

Endel Tulving. Episodic and semantic memory. In Endel Tulving and Wayne Donaldson (eds.), *Organization of Memory*, pp. 381–403. Academic Press, New York, 1972.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Shuo Yang, Caren Han, Siwen Luo, and Eduard Hovy. MAGIC-VQA: Multimodal and grounded inference with commonsense knowledge for visual question answering. In *Findings of the Association for Computational Linguistics: ACL*, pp. 16967–16986, 2025a.

Zhen Yang, Zhuo Tao, Qi Chen, Liang Li, Yuankai Qi, Anton Van Den Hengel, and Qingming Huang. Separation of powers: On segregating knowledge from observation in LLM-enabled knowledge-based visual question answering. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24753–24762, 2025b. doi: 10.1109/CVPR52734.2025.02305.

Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. An empirical study of GPT-3 for few-shot knowledge-based VQA. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. LLaMA-Adapter: Efficient fine-tuning of large language models with zero-initialized attention. In *International Conference on Learning Representations (ICLR)*, 2024a.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *Transactions on Machine Learning Research (TMLR)*, 2024b.

A Supplementary Methodology

A.1 Task Definition and Notation

Given an image I and question Q, KB-VQA combines visual information with external knowledge to produce an answer $\hat{A}$. The question input includes answer options and background text permitted by the evaluation protocol. For ScienceQA, this includes lecture annotations but excludes reference rationales. Reference-answer feedback is never part of the current question input. All model parameters remain fixed.

The frozen fact reasoner $\mathcal{M}$ first produces a global image description C. It parses this description and the question into open-vocabulary visual prompts and knowledge entities:

$$(\mathcal{P}_1, \mathcal{E}_1) = f_{\mathrm{p}}(Q, C). \tag{6}$$

where, $\mathcal{P}_1 = \{p_{1,j}\}_{j=1}^{m}$ contains textual descriptions of target objects. The set $\mathcal{E}_1$ contains entities for knowledge retrieval. Task subscripts on f distinguish calls to the same frozen model: f_{p} parses the question, f_{v} verifies visual candidates, f_{r} assesses readiness, and f_{m} refines memories. Answer generation also uses the frozen model. These functions do not introduce separately trained models.

The index t identifies an EDAR iteration within one question. The index i identifies a question across the task stream. The actual stopping iteration is τ for a single question and τ_i for question Q_i. The sets $\mathcal{V}_t$, $\mathcal{K}_t$, and $\mathcal{W}_t$ contain visual evidence, accumulated graph knowledge, and validated

program outputs. Graph knowledge starts empty before the first retrieval. The record $\mathcal{H}_{t-1}$ contains reasoning and actions from earlier iterations. The complete record Π_i preserves the reasoning and actions through answer generation. It supports memory feedback only after the answer is produced.

The superscript (s) identifies a completed memory version. Question Q_i reads $\mathcal{B}^{(s_i)}$ and selects a fixed memory set $\mathcal{U}_i$. For the same question, $\mathcal{U}_Q$ denotes this set in the single-question reasoning procedure. A background update may begin from a later version $\mathcal{B}^{(s)}$. It does not change the memory context already selected for an ongoing question.

A.2 Core Prompt Templates

Table 7 presents visual candidate verification, Table 8 covers evidence arbitration and action selection, and Table 9 describes feedback-based memory refinement. These core prompt templates are condensed for presentation; braces denote question-specific inputs.

Table 7: Visual candidate verification prompt.

Visual Candidate Verification

Look at this image crop. Does it primarily show a '{target object}'? Reply ONLY with a valid JSON block, nothing else.

Format: {`"is_match": true, "confidence_score": 0.9`}.

Table 8: Evidence arbitration and action selection prompt.

Evidence Arbitration and Action Selection

You are a Scientific Reasoning Adjudicator. Answer by weighing evidence from multiple agents.

Question and choices: {question, choices}.
Evidence: {visual observations, verified candidates, external knowledge, relevant memories}.
Evaluator feedback and reasoning history: {feedback, history}.

Task 1: Resolve disagreements between VLM observations and SAM3 detections. Check counts because SAM3 can miss objects. For abstract concepts and relationships, rely on the VLM.

Task 2: Use evaluator feedback to choose among answering, focused visual rechecking, and further knowledge retrieval. Request visual rechecking for conflicting or missing visual details, and knowledge retrieval for missing facts or definitions.

Task 3: When instructed to commit, provide the final answer. Return your answer or requested action with a brief justification.

A.3 Co-verified Visual Grounding

Candidate generation and semantic verification. SAM3 receives the open-vocabulary visual prompts and proposes candidate regions. For prompt $p_{t,j}$, candidate k has mask $m_{t,j,k}$ and bounding box $b_{t,j,k}$. Equation 1 applies f_{v} to the masked crop and its prompt. The result $a_{t,j,k}$ records whether the region matches the target concept. The score $v_{t,j,k}$ records model-reported confidence. The set $\mathcal{A}_{t,j}$ retains all candidates and their annotations, including candidates with $a_{t,j,k} = 0$.

Comparison with the whole image. After candidate verification, $\mathcal{M}$ compares the annotated candidates with observations of the whole image. This step checks for disagreements in object identity, attributes, counts, and relations. It is a separate comparison, not a second output of Equation 1. Candidate records and whole-image observations jointly form $\mathcal{V}_t$. Local and global observations provide complementary context.

Table 9: Feedback-based memory refinement prompt.

Feedback-Based Memory Refinement

You are the meta-cognitive reflection module. A previously stored memory has failed. Analyse and upgrade it.

Existing memory: {memory}.
Failed case: {question, choices, prediction, reference answer}.
Original reasoning history: {reasoning history}.

Task 1: Diagnose the failure as a strategy, computation, knowledge, or perception gap.

Task 2: Explain what the old memory missed and revise its reusable guidance. The revised steps should subsume the previous advice.

Task 3: For a computation gap, improve the existing executable procedure or replace a repeatedly failing textual strategy with a procedure.

Generalise: avoid copying unique numbers or proper nouns from this case. Return a short title, a description of the correction, and the revised guidance or executable procedure.

Using disagreement during reasoning. Failed candidates remain available with their failure annotations. Disagreements between local and global observations also remain available to the reasoner. The reasoner considers these records alongside relevance, confidence, and visual context. An empty candidate set indicates missing perceptual support rather than proof of object absence. Verification provides an assessment. Unresolved visual issues can trigger rechecking of the image with existing segmentation results as prompts. Unchanged prompts, candidates, and annotations are reused. Segmentation and candidate verification need not run again at every iteration.

A.4 Memory Representations and Illustrative Entries

The initial memory bank contains 420 entries: 296 Episodic Memory entries, 53 Semantic Memory entries, and 71 Procedural Memory entries. The serialised TYPE values `Episodic`, `Semantic_Rule`, and `Tool` implement these three conceptual categories, respectively. These counts describe the initial bank state rather than a fixed architectural capacity. The entries below are independent examples of representation and use.

Table 10: Representative entries from the initial memory bank. Text is shortened for readability, while titles and statistics are retained.

Memory type	Stored title	Content and usage
Episodic Memory	Verify consistency between calculated values and options	Case-linked guidance to recompute requested quantities before matching an option. Its provenance remains episodic; recorded usage outcomes include one success and one failure.
Semantic Memory	Odds ratio calculation from 2×2 contingency tables	Generalized odds-ratio knowledge, $\mathrm{OR} = ad/(bc)$, where (a, b) and (c, d) are case/control counts in exposed and unexposed groups. Recorded usage outcomes include seven successes and two failures.
Procedural Memory	Compass traverse angle computation	Executable operations for parsing degree-minute bearings and computing normalized interior angles. Its zero-success, five-failure history motivates reassessment of applicability and content.

Episodic Memory preserves reasoning experiences, error analyses, and corrective lessons from specific tasks. An Episodic Memory record contains MEMORY_ID, TYPE, TASK_QUERY, ORIGINAL_TRAJECTORY_REFS, ORIGINAL_TRAJECTORY_SUMMARY, MEMORY_ITEMS, STATS, and a CONSOLIDATED flag. Each item contains a TITLE, diagnostic DESCRIPTION, and repair CONTENT; statistics include success, failure, and last-update fields. Semantic Memory is formed by clustering similar episodic memories and summarizing each cluster into reusable rules. It retains links to the source episodes and stores their corresponding reasoning steps in a dedicated field. The original

episodic memories remain available after abstraction. Later reasoning can therefore use both general rules and concrete experience. Procedural Memory additionally stores a function name, parameter schema, and executable code. Usage statistics record observed outcomes associated with retrieval or execution. An episodic lesson may be phrased as general advice while remaining linked to a particular case. Conversely, semantic guidance may describe calculation steps without being an executable procedure. Provenance, abstraction, and inference-time use jointly determine these distinctions.

Semantic Memory reuse across questions. For example, the memory bank contains the Semantic Memory entry *Strictly Separate Visual Evidence from Historical Inference*. This entry distinguishes observable chart content from background knowledge. For an MMMU question about British economic interactions with India, the recorded trace retrieves this entry and notes that the GDP table does not identify commodities. It then combines the table's temporal context with historical knowledge to select *Cotton*, matching the annotation. Another trace retrieves the same entry for a question about world silver flows. It combines the depicted trade routes with historical context to select the annotated answer. These cases illustrate reuse of the evidence–knowledge distinction across different scenarios and questions. They highlight the value of Semantic Memory in preserving reasoning principles that generalize beyond individual cases and guide later questions. This highlights the value of semantic memory in preserving generalizable reasoning principles and applying them across different scenarios.

A.5 Memory Formation, Refinement, and Retrieval

SHM stores declarative memory as $\mathcal{B}_{\mathrm{D}}^{(s)} = \mathcal{B}_{\mathrm{E}}^{(s)} \cup \mathcal{B}_{\mathrm{S}}^{(s)}$ and procedural memory as $\mathcal{B}_{\mathrm{P}}^{(s)}$. The subscripts identify episodic, semantic, and procedural entries. All three types remain retrievable. Episodic entries preserve concrete reasoning processes, error analyses, and corrective lessons. Semantic entries summarize related episodes into reusable rules. Procedural entries provide executable methods with explicit applicability conditions. Appendix A.4 describes their stored content and examples.

Cold-start construction. The system first runs EDAR on an initial set of training samples with memory retrieval disabled. It records the reasoning trajectories and obtains feedback from reference answers after each prediction. Incorrect answers and correct answers reached after multiple iterations can supply reusable experience. Valid extracted entries are organized by type. Related episodes support semantic abstraction, and recurring computational needs can support procedural memory. These steps produce the initial completed bank $\mathcal{B}^{(0)}$ without updating model parameters. Memory updates remain enabled during evaluation. After the current prediction is recorded, reference-answer feedback updates usage statistics and refines relevant entries. Completed revisions are available to later evaluation questions. The reference answer for the current question is not available when its prediction is generated. These results therefore evaluate reasoning with feedback-driven memory updates, while all model parameters remain fixed.

Retrieval by title similarity. For question Q_i, the system reads $\mathcal{B}^{(s_i)}$. The text encoder ϕ maps the question to embedding $\mathbf{q}_i$ and the title of each memory entry μ_ℓ to embedding $\mathbf{z}_\ell$. The encoder receives the entry's title, not its full content. Let $\sigma_i(\mu_\ell)$ denote the cosine similarity between these two embeddings. The initial candidate set is

$$\mathcal{C}_i = \mathrm{Top}_{k'_{\mathrm{m}}}\left(\mathcal{B}^{(s_i)}; \sigma_i\right), \qquad k'_{\mathrm{m}} = \max(3k_{\mathrm{m}}, 5). \tag{7}$$

The operator returns up to k'_{m} entries ranked by title similarity; if the bank contains fewer entries, all available entries are returned. The candidate budget expands the pool before reranking. The constants 3 and 5 specify the existing retrieval budget; no optimality claim is attached to them. Figure 6 reports sensitivity to the final retrieval count k_{m}.

The ranking score $w_{i\ell}$ combines title similarity, past usage performance, and memory type. The cosine similarity is multiplied by the usage weight $\rho_\ell \in [0, 1]$ and type weight β_ℓ. Entries with zero usage weight are excluded before thresholding. The type weight is 1 for episodic memory, 1.15 for semantic memory, and 1.20 for procedural memory. These settings give semantic and procedural entries a 15% and 20% boost relative to episodic entries with the same similarity and usage weight.

The final retrieval step applies the threshold θ_{m} to the weighted score $w_{i\ell}$ and selects up to k_{m} eligible entries. The result can be empty. The selected set $\mathcal{U}_i$, also denoted by $\mathcal{U}_Q$, stays fixed

Algorithm 1 SHM Refinement and Hierarchical Update after Answering.

Require: Completed case $(Q_i, \Pi_i, \hat{A}_i, y_i, c_i, \tau_i)$; retrieved set $\mathcal{U}_i$; current write version $\mathcal{B}^{(s)}$
Ensure: Completed memory version $\mathcal{B}^{(s+1)}$
Initialize $\mathcal{F}_i, \mathcal{F}'_i, \mathcal{N}_i \leftarrow \varnothing$
Locate retrieved entries in the current version by their identifiers
if $c_i = 1$ **then**
    Reinforce usage statistics of relevant entries
else
    Update usage statistics and assess applicability of retrieved entries
    for all applicable retrieved entries μ_ℓ that need revision **do**
        $\mu'_\ell \leftarrow f_{\mathrm{m}}(\mu_\ell, Q_i, \Pi_i, \hat{A}_i, y_i)$
        if the revision is valid **then**
            Add μ_ℓ to $\mathcal{F}_i$ and μ'_ℓ to $\mathcal{F}'_i$
        end if
    end for
    Adjust retrieval priority for inapplicable entries without rewriting them from this case
end if
if $\eta_i = 1$ under Eq. 3 **then**
    Extract reusable experience from the completed case and feedback
    Add valid new memories to $\mathcal{N}_i$
end if
Apply the content update and hierarchical organization in Eq. 4
Retain source episodes and their links when forming semantic rules
Make the completed version available to later questions
return $\mathcal{B}^{(s+1)}$

throughout the current question. The threshold θ_{m} controls memory selection and is distinct from the readiness threshold θ_{r}. Declarative memory provides knowledge and concrete experience. Retrieved procedures become available for execution when the question is applicable and all required inputs are present.

Feedback and refinement of existing memory. Reference answer y_i is used only after the prediction $\hat{A}_i$ is produced. The task protocol supplies correctness $c_i \in \{0, 1\}$. A correct answer reinforces usage statistics for relevant memories. An incorrect answer prompts examination of the retrieved memories. An applicable memory with incomplete or incorrect guidance is refined by f_{m} in Equation 2. Refinement can correct the guidance, add applicability conditions, or revise a program. A valid revision replaces the corresponding entry by identifier. If no valid revision is produced, the original memory remains.

A memory can be retrieved because its title is similar to the question but still be inapplicable. Such a case calls for adjustment of its retrieval priority, not automatic rewriting of its content. Usage statistics record outcomes associated with memory use. Refining existing memories reduces redundant additions; it does not impose a fixed upper bound on memory size.

Selective formation of new memories. Equation 3 requires both an empty retrieved set $\mathcal{U}_i$ and an informative outcome: an incorrect answer or a correct answer reached after multiple iterations. The indicator $\eta_i = 1$ permits an attempt to extract new experience, not unconditional insertion. The extracted content must be valid before it is added. Reflection compares the prediction and reasoning record with the reference answer. It identifies the difficulty and a reusable correction. An incorrect prediction remains part of the failure context and is not promoted to a general fact. The title identifies the lesson, the description explains the difficulty, and the content records the guidance. When relevant memories already exist, the system prioritizes refinement over duplicate additions.

Abstraction across related episodes. The system groups episodic entries by shared capability gaps and similar solution patterns. Semantic similarity in their error descriptions and repair content supports this grouping. A shared topic alone does not establish a common solution. When a coherent group contains at least N unconsolidated entries, the reasoner summarizes the group into reusable semantic rules. Incidental numbers, names, and answer choices are omitted unless they define the

Algorithm 2 EDAR reasoning and post-answer SHM feedback in SAGE-VQA.

Require: Image I; question Q; frozen reasoner $\mathcal{M}$; perception enhancer $\mathcal{S}_{\text{seg}}$; memory version $\mathcal{B}^{(s_i)}$; threshold θ_{r}; budget T

Ensure: Answer $\hat{A}$ and complete reasoning record Π

Generate global image description C; set $(\mathcal{P}_1, \mathcal{E}_1) \leftarrow f_{\text{p}}(Q, C)$
Retrieve fixed memory set $\mathcal{U}_Q$ using Appendix A.5
$\mathcal{K}_0 \leftarrow \varnothing$; $\mathcal{W}_0 \leftarrow \varnothing$; $\mathcal{H}_0 \leftarrow \varnothing$
Execute applicable procedures with already available inputs; retain validated outputs in $\mathcal{W}_0$
for $t = 1$ **to** T **do**
    if $t = 1$ **then**
        Generate candidates; verify them with Eq. 1; compare with whole-image observations
        Form $\mathcal{V}_1$ and retrieve initial graph knowledge using Eq. 5
    else
        Reuse prior evidence and perform the selected visual recheck or knowledge query
        Update $\mathcal{V}_t$ and $\mathcal{K}_t$ as needed; retain unchanged information
    end if
    Initialize $\mathcal{W}_t \leftarrow \mathcal{W}_{t-1}$, retaining only outputs valid for the current inputs
    Execute newly applicable procedures; add validated outputs to $\mathcal{W}_t$
    Reason with Q, $\mathcal{V}_t$, $\mathcal{K}_t$, $\mathcal{U}_Q$, $\mathcal{W}_t$, and $\mathcal{H}_{t-1}$
    Append current reasoning and actions to $\mathcal{H}_{t-1}$ to form $\mathcal{H}_t$
    Form $\mathcal{X}_t$ from the question, visual evidence, graph knowledge, memories, program outputs, and reasoning record
    $(r_t, \mathcal{D}_t) \leftarrow f_{\text{r}}(\mathcal{X}_t)$
    if $r_t \geq \theta_{\text{r}}$ **or** $t = T$ **then**
        $\tau \leftarrow t$; **break**
    end if
    Select visual rechecking or knowledge supplementation using $\mathcal{D}_t$ and $\mathcal{H}_t$
end for
Generate $\hat{A}$ from current evidence, memories, program outputs, and reasoning history; finalize complete record Π
if memory updates are enabled and reference answer y becomes available **then**
    Compute correctness c under the task protocol
    Submit $(Q, \Pi, \hat{A}, y, c, \tau)$ and retrieved memory identifiers to Algorithm 1 asynchronously
end if
return $(\hat{A}, \Pi)$

rule's applicability. Each semantic entry retains links to its source episodes and stores their corresponding reasoning steps. The original episodes remain available after abstraction. Later reasoning can therefore use both a general rule and concrete steps from its source cases. No new semantic entry is formed when the grouping conditions are unmet.

Procedural memory and execution. Recurring computational needs can motivate an executable procedure or refinement of an existing one (Luo et al., 2026). Construction checks the generated code and trial execution. Detected failures prompt revision. At inference time, retrieval alone does not trigger execution. The current question must fit the procedure's applicability conditions, and its required inputs must be available. Inputs can come from the question, image descriptions, or acquired visual information. Only outputs that pass execution and output checks enter $\mathcal{W}_t$. A reusable formula can remain semantic guidance; procedural memory stores and invokes an executable implementation.

A.6 EDAR Reasoning and SHM Feedback Procedure

Algorithm 2 expands the EDAR strategy in Section 3.4 and shows its post-answer handoff to SHM. Memory retrieval runs once per question. Each iteration reuses the current evidence and validated program outputs unless an action updates them. Procedural inputs can come from the question, image descriptions, or acquired visual information. Execution depends on input availability, not on a mandatory position after segmentation.

Asynchronous update and version use. Algorithm 1 summarizes post-answer updates. The current write version s need not equal the read version s_i. Updates locate memories by their identifiers. Only current-version memories with valid replacements enter $\mathcal{F}_i$, matched by identifier. Their revisions form $\mathcal{F}_i'$, and valid new memories form $\mathcal{N}_i$. Equation 4 replaces the corresponding old memories $\mathcal{F}_i$, adds new content, and applies hierarchical organization h. This equation describes content replacement and formation; the usage-statistic and retrieval-priority adjustments in Algorithm 1 are also retained in the completed version. Completed versions become available to later questions. They do not overwrite $\mathcal{U}_i$ for an ongoing question. The main evaluation follows this update path after each answered question; valid content revisions are applied when the refinement or formation conditions are met.

Computational cost. Each question performs one memory retrieval. In an iteration that generates and verifies all candidates, the visual stage uses $O(mK)$ candidate verification calls. Here, m is the number of active prompts and K bounds the candidates per prompt. Structural and visual retrieval use $O(n+mK)$ queries, where n is the number of knowledge entities. These are full-refresh bounds; later iterations may reuse earlier results and perform fewer calls. Each iteration also includes whole-image comparison as needed and one readiness assessment. Procedural execution costs depend on the retrieved program. Post-answer memory updates run asynchronously and do not block answer generation.

A.7 Hybrid Multimodal Knowledge Retrieval

Knowledge retrieval is part of EDAR in Section 3.4. The structural channel queries a Neo4j-hosted MMKG using parsed entities (Liu et al., 2019). The visual channel embeds locally verified image regions with CLIP (Radford et al., 2021) and searches the graph's image index with FAISS (Johnson et al., 2019). Using the candidate records defined in Section 3.2, the image regions for visual retrieval are

$$\mathcal{O}_t = \bigcup_j \{I[b_{t,j,k}] : (m_{t,j,k}, b_{t,j,k}, a_{t,j,k}, v_{t,j,k}) \in \mathcal{A}_{t,j},\ a_{t,j,k} = 1\}\,. \tag{8}$$

The mask crop is used for candidate verification, while the bounding-box crop is used for visual retrieval. Selecting locally verified regions for retrieval does not remove failed candidates from $\mathcal{V}_t$.

Let ϕ_I be the image encoder and $\mathcal{Z}_I$ the indexed image embeddings. The operator NN_2 returns up to two indexed image identifiers nearest to a query embedding. For a set of image identifiers, ζ returns the union of their linked graph entities. An empty input yields an empty entity set. The visually matched entity set is

$$\mathcal{E}_t^{\mathrm{vis}} = \bigcup_{o \in \mathcal{O}_t} \zeta(\mathrm{NN}_2(\phi_I(o); \mathcal{Z}_I))\,. \tag{9}$$

Both the parsed entities and visually matched entities support graph retrieval. Let Γ return the selected one-hop relations and facts under the per-entity retrieval budget. The retrieval function in Equation 5 is

$$g(\mathcal{E}_t, \mathcal{O}_t) = \Gamma\big(\mathcal{E}_t \cup \mathcal{E}_t^{\mathrm{vis}}\big)\,. \tag{10}$$

Equation 5 adds this result to $\mathcal{K}_{t-1}$ rather than replacing earlier knowledge. The initial retrieval uses both channels. Later queries target the missing information identified during reasoning. If no new query is issued or retrieval returns no new facts, the existing knowledge is retained. Appendix D reports the retrieval budgets.

A.8 Readiness Assessment and Bounded Reasoning

At iteration t, the evaluator receives $\mathcal{X}_t$, which contains the question, visual evidence, graph knowledge, relevant memories, available program outputs, and the reasoning record. The readiness assessment returns a score r_t and unresolved issues $\mathcal{D}_t$. Each issue includes a category and an explanation. Breadth identifies a missing aspect of the question. Depth identifies insufficient detail. Contradiction identifies incompatible observations or claims. Relevance identifies information outside the question's scope. These descriptions guide the next action. Visual uncertainty motivates rechecking, while missing external knowledge motivates further retrieval.

Readiness estimates whether the available information supports an answer. It is not a calibrated probability that the answer is correct. The stopping iteration τ is the first iteration at which $r_t \geq \theta_{\mathrm{r}}$,

or the maximum budget T if the threshold is never reached. The system then answers using the available evidence, memories, program outputs, and reasoning history. Budget exhaustion does not establish evidence sufficiency. Missing image details and limited graph coverage can leave uncertainty unresolved.

B EXAMPLE FROM THE MICROQUANT-VQA BENCHMARK

Figure 5 shows a representative instance from the MicroQuant-VQA benchmark, drawn from the lineal-path question family.

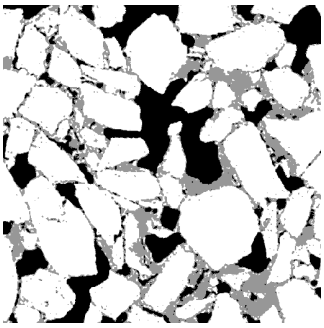


Figure 5: A 512×512 three-phase segmented micro-CT slice (voxel $= 1.6854\,\mu$m/px) from the MicroQuant-VQA benchmark. Voxels are labeled as void (black, pixel 0), biopolymer (gray, pixel 151), or aggregate (white, pixel 255). The question asks for the lineal-path probability $L(r)$ of the aggregate phase at $r = 64.01\mu$m.

Question. This image is one slice of a three-phase segmented micro-CT volume (voxel $=$ 1.6854 µm/px). In this image: void $=$ black (pixel 0), biopolymer $=$ gray (pixel 151), aggregate $=$ white (pixel 255). The lineal-path function $L(r)$ of a phase is the probability that a randomly placed segment of length r, with both ends in the image, lies entirely within that phase. For the aggregate phase at $r = 64.01\,\mu$m, what is the lineal-path probability?

Options. (A) 0.217 (B) 0.2472 (C) 0.0124 (D) 0.147

Ground-truth answer. (A) 0.217 (full-precision value: 0.216975740131578).

Why this task is difficult for frozen VLMs. The near-chance performance of the tested Qwen3.5 backbones—23.10% for 9B and 29.30% for 27B, against a 25% random-guess baseline—reflects three structural challenges. First, the input is an abstract three-label segmentation mask rather than a natural image. Second, $L(r)$ requires explicit spatial measurement, which the tested models did not recover reliably by inspection. Third, every distractor is a valid same-phase value at another distance, weakening surface heuristics. Together, these properties make MicroQuant-VQA a diagnostic probe of the perception–computation gap. In our setting, grounding the target phase, computing $L(r)$ from the delineated region, and checking the requested distance were necessary for the observed improvement.

C DATASET DETAILS

Table 11 summarises the evaluated splits and task formats.

ScienceQA. ScienceQA (Lu et al., 2022) contains 21,208 questions spanning natural science, social science, and language science. Questions may include text, diagrams, or natural images. Lecture annotations are supplied as background knowledge during inference; reference rationale annotations are excluded. We evaluate the official test split and use top-1 multiple-choice accuracy. The diverse contexts make ScienceQA suitable for testing grounded perception and compositional reasoning.

A-OKVQA. A-OKVQA (Schwenk et al., 2022) contains 24,903 questions grounded in COCO images. Its official split contains 17,056 training, 1,145 validation, and 6,702 test questions. Each

Table 11: Statistics and evaluation protocols for the four benchmarks.

Dataset	Available questions	Evaluated split	Answer format
ScienceQA	21,208	official test	multiple choice
A-OKVQA	24,903	validation (1,145)	MC and direct answer
MMMU	11,500	validation (900)	multiple choice and open-ended
MicroQuant-VQA	4,970	validation (1,000)	multiple choice

Table 12: Implementation settings used for all evaluations unless noted otherwise.

Component	Hyperparameter	Setting
Fact-reasoner decoding	temperature	0.1
EDAR	maximum iterations T	3
EDAR	readiness gate θ_{r}	0.95
Memory retrieval	entries per query k_{m}	2
Memory retrieval	candidate budget k'_{m}	$\max(3k_{\mathrm{m}}, 5)$
Memory retrieval	episodic/semantic/procedural type weights	1/1.15/1.20
Semantic consolidation	minimum coherent-group size N	10
MMKG retrieval	triplets per entity k_{g}	15
FAISS retrieval	neighbours per locally supported crop	2

question has four choices and ten free-form answers. We evaluate both formats on validation. Direct answers follow the official VQAv2 protocol: predictions and annotations receive the standard punctuation, digit, article, and contraction normalization, followed by leave-one-out soft accuracy with exact normalized matches.

MMMU. MMMU (Yue et al., 2024) contains approximately 11,500 college-level questions across 30 subjects and 183 subfields. It covers six disciplines and 30 heterogeneous image types, including charts, diagrams, tables, maps, and chemical structures. We evaluate all 900 validation questions. Evaluation follows the official code: option labels are parsed for multiple-choice items; open responses are reduced to candidate strings and numbers, strings use normalized containment, and numbers use equality after two-decimal normalization.

MicroQuant-VQA. MicroQuant-VQA evaluates cross-domain transfer to quantitative reasoning in materials science. It is derived from three-phase segmented micro-CT slices of biopolymer-bound soil composites (Miao et al., 2024). Each 512×512 image labels void, biopolymer, and aggregate at $1.6854\,\mu\mathrm{m}$ per pixel. Questions require computing the lineal-path probability of a specified phase at a given distance (Torquato, 2002). Each question has four options, with distractors drawn from valid same-phase values at other distances. We evaluate 1,000 validation questions and report top-1 multiple-choice accuracy. This benchmark complements the preceding three by focusing on numerical measurement from segmented microstructures, probing whether grounded reasoning and reusable computational procedures transfer to a specialised scientific domain.

D Implementation and Hyperparameters

Table 12 reports the shared inference and memory settings. Decoding is fixed across datasets and backbone sizes. Model parameters remain fixed, while memory entries and usage statistics can change through post-answer feedback during evaluation. Retrieval budgets were selected on the ScienceQA training split and fixed before evaluation; the sensitivity sweeps in Figure 6 retain the main-table configuration apart from the budget being varied. We use $\theta_{\mathrm{r}} = 0.95$ throughout. Candidate verification returns a binary semantic match and model-reported confidence. Whole-image comparison records disagreements. Neither check introduces an additional confidence threshold in the reported procedure. The memory selection threshold θ_{m} is distinct from the readiness threshold θ_{r}.

The fact reasoner receives the image, question input, annotated evidence, type-aware memory entries, and available validated procedural outputs. When EDAR is enabled, SAM3 receives prompts

derived from the question and initial caption. Each candidate enters the reasoning state with its semantic judgment; failed candidates retain their failure flags. Retrieved procedures execute when their applicability conditions and input requirements are met. Inputs may come from the question, image descriptions, or acquired visual information. Only validated outputs enter the reasoning context.

E Details of the Comparative Methods

Following the distinction in the Introduction, we group comparative methods into training-based and training-free approaches according to whether any framework component requires additional parameter training or adaptation.

Reporting protocol. Published scores follow the cited evaluation sources. "TF" denotes a framework requiring no additional parameter training or adaptation in any component. Off-the-shelf pretrained models are used with fixed weights throughout framework construction and inference. Our bare-backbone and incremental comparisons use identical evaluation code. QACap's occurrence-based direct-answer results remain excluded because they differ from the official VQAv2 scoring protocol used here.

Training-based methods. These methods develop reasoning capabilities through backbone fine-tuning, reinforcement learning, parameter-efficient adaptation, or auxiliary-component training. They require parameter optimisation even when the answer model remains frozen.

- **ViLBERT** uses co-attention between visual-region and language streams. The A-OKVQA baseline fine-tunes this pretrained model on the target dataset (Lu et al., 2019; Schwenk et al., 2022).
- **GPV-2** expands a general-purpose vision model's concept coverage with web supervision; the reported A-OKVQA configuration is subsequently fine-tuned for answering (Kamath et al., 2022; Schwenk et al., 2022).
- **Multimodal-CoT** trains rationale generation and answer inference as two stages, integrating textual and visual features so that generated rationales condition answer prediction (Zhang et al., 2024b).
- **LLaVA and LLaVA+GPT-4 judge** use a LLaVA model fine-tuned on ScienceQA. In the ensemble, disagreements between LLaVA and text-only GPT-4 trigger a further GPT-4 call to adjudicate the final answer (Liu et al., 2023).
- **CalibRL** trains Qwen2.5-VL-7B with hybrid-policy reinforcement learning using verifiable rewards, distribution-aware advantage weighting, and expert-guided calibration. Training uses geometry examples from ViRL39K; ScienceQA and MMMU are reported as out-of-domain evaluations (Huang et al., 2026).
- **LLaMA-Adapter** trains visual projections and adaptation prompts with zero-initialized attention gating while freezing the image encoder and LLaMA (Zhang et al., 2024a).
- **LaVIN** trains mixture-of-modality adapters and routing mechanisms for visual–language instruction tuning with frozen pretrained components (Luo et al., 2023).
- **MR-MKG** trains a relational graph attention network and visual/knowledge adaptation modules while freezing the LLM and visual encoder. Training includes MMKG-grounded pretraining and task-specific adaptation. The ScienceQA result uses FLAN-UL2-19B, with 248M trainable parameters reported for the auxiliary components (Lee et al., 2024).
- **RG-VQA** trains a ColBERTv2 retriever to select relevant knowledge-graph triples and supplies them as generator context. The reported Bunny-8B configuration additionally uses LoRA fine-tuning; it is a retriever–generator pipeline (Sravanthi et al., 2025).
- **PromptCap** fine-tunes a 470M OFA captioner on GPT-3-synthesized, VQAv2-derived examples. It then prompts frozen GPT-3 with captions and in-context examples for A-OKVQA. The paper reports the code-davinci-002 model as 175B. Its validation direct-answer score is 56.3 under VQAv2 soft accuracy (Hu et al., 2023a).

- **QACap** trains a question-aware captioner with feature distillation and regional consistency learning on captions derived from KB-VQA training splits, including A-OKVQA, and passes the resulting visual descriptions to a frozen LLM (Yang et al., 2025b).
- **MAGIC-VQA** combines commonsense retrieval and type-based filtering with a trained GCN/MLP that supplies answer-confidence signals to a frozen LVLM. The paper reports 0.33M trainable parameters for this auxiliary module. The A-OKVQA and MMMU scores in our tables use GPT-4o-mini; the paper does not specify its parameter count. The 0.33M figure therefore excludes the backbone (Yang et al., 2025a).
- **V3Fusion** selects ensembles of pretrained VLMs and trains a fusion module for their outputs. Multiple-choice tasks use an MLP; open-ended tasks use a trained encoder–decoder. The A-OKVQA ensemble contains LLaVA-v1.6-Vicuna-13B, LLaVA-v1.6-Vicuna-7B, and DeepSeek-VL2-Tiny. The MMMU ensemble replaces the 13B LLaVA with Qwen2.5-VL-7B. The reported results include uncertainty-based rectification. We use the Table 2 scores of 89.17 on A-OKVQA and 56.09 on MMMU (Tekin et al., 2026).

Training-free methods. These methods use pretrained models with fixed parameters, either through direct inference or through prompting, retrieval, and tool-based composition.

- **MG^2-RAG** builds a multimodal graph linking entities, visual regions, and document chunks, then propagates query relevance to retrieve evidence for a frozen answer model (Dai et al., 2026).
- **VLMs-only** use off-the-shelf GPT-4V, GPT-4o-mini, Qwen2.5-VL-7B, Gemini-3.1-Pro, and GPT-5.2 without framework-specific retrieval or adaptation. Gemini-3.1-Pro and GPT-5.2 ScienceQA scores are taken from the direct-inference evaluations reported by MG^2-RAG (Yue et al., 2024; OpenAI, 2024; Bai et al., 2025; Google DeepMind Team, 2026; OpenAI Team, 2026; Dai et al., 2026).

Controlled internal baselines. The bare-VLM baseline uses the frozen fact reasoner without retrieval, SAM3, or memory. The incremental ablation first enables MMKG retrieval alone. The next stage completes EDAR by adding visual verification and adaptive reasoning to this retrieval baseline. The final stage adds SHM. Thus, the EDAR row measures the increment over its retrieval-only baseline. MMKG retrieval remains part of the complete strategy.

F Additional Experimental Analysis

F.1 MicroQuant-VQA Transfer

Table 13: Accuracy on MicroQuant-VQA validation sets.

Configuration	Accuracy
Bare VLM (9B)	23.10
SAGE-VQA (9B)	35.80 (+12.70)
Bare VLM (27B)	29.30
SAGE-VQA (27B)	**87.50** (+58.20)

F.2 Qwen3.5-27B Component Ablation

We repeat the incremental ablation with Qwen3.5-27B to examine whether the framework's components remain useful with a stronger fact reasoner and how their contributions change relative to the 9B setting in Table 4.

Table 14 follows the same staged comparison. MMKG retrieval supplies external relations and facts. Completing EDAR adds visual verification and adaptive evidence acquisition. The final stage enables use and post-answer refinement of declarative and procedural memories. Each stage improves both benchmarks under this ordering. The EDAR increment is measured over retrieval alone, while the SHM increment is measured over the complete EDAR strategy.

Table 14: Component contributions on Qwen3.5-27B. The +EDAR row adds visual verification and adaptive reasoning to MMKG retrieval, as in Table 4. Parentheses report gains over the preceding configuration.

Configuration	ScienceQA (%)	MMMU (%)
Bare VLM (Qwen3.5-27B)	91.98	54.20
+ MMKG retrieval	93.00 (+1.02)	58.36 (+4.16)
+ EDAR (full strategy)	96.11 (+3.11)	68.12 (+9.76)
+ SHM (full SAGE-VQA)	**98.56** (+2.45)	**75.32** (+7.20)

Comparison with the 9B ablation reveals benchmark-dependent scale effects. On ScienceQA, the higher bare-27B accuracy leaves less room for improvement, consistent with the smaller absolute component gains. On MMMU, however, the gain from completing EDAR beyond retrieval alone increases from 6.85 to 9.76 points, and the memory gain from 5.01 to 7.20, while the retrieval gain remains comparable. This pattern suggests that a stronger backbone can benefit more from grounded evidence assessment and experience reuse on specialized reasoning tasks. Scaling therefore remains complementary to the framework's evidence and memory mechanisms.

F.3 Complementarity of Declarative and Procedural Memory

This ablation examines whether declarative and procedural memory make complementary contributions to SAGE-VQA. We group Episodic and Semantic Memory as declarative memory: they provide case-specific experience and generalized knowledge, respectively. Procedural Memory instead supplies executable operations for applying knowledge to the current problem. We compare no memory retrieval, declarative retrieval, and declarative plus procedural retrieval. Each configuration starts from the same initial bank and updates its own copy from post-answer feedback during evaluation. The comparison measures the effect of enabling different memory types together with their subsequent memory updates.

Table 15: Contributions of declarative and procedural memory on Qwen3.5-9B. Episodic and Semantic Memory jointly form the declarative category; adding Procedural Memory tests its incremental value beyond declarative retrieval. Parentheses report gains over the preceding row. All configurations start from the same initial bank and update their memories separately during evaluation.

Memory configuration	ScienceQA (%)	MMMU (%)
No retrieved memory	92.76	61.29
+ Episodic + Semantic Memory	94.28 (+1.52)	63.21 (+1.92)
+ Episodic + Semantic + Procedural Memory	**95.33** (+1.05)	**66.30** (+3.09)

Table 15 shows higher scores with declarative retrieval and further gains with procedural retrieval. Each configuration uses its own evolving memory bank. The gains therefore reflect memory reuse and subsequent updates, rather than retrieval from identical fixed contents. Declarative memory provides prior experience and generalized knowledge, while procedural memory supplies reusable computations. The larger gain from adding procedural memory on MMMU is consistent with its specialized computational demands.

F.4 Retrieval-Budget Sensitivity

Figure 6 reveals a non-monotonic relationship between retrieval budget and answer accuracy. Within the tested ranges, performance peaks under the configuration $k_{\mathrm{m}} = 2$ and $k_{\mathrm{g}} = 15$, then declines when either budget is increased further. Thus, retrieving more memory entries or MMKG triplets does not necessarily improve reasoning. This pattern is consistent with a trade-off between evidence coverage and contextual selectivity: broader retrieval may introduce redundant or weakly relevant material that dilutes useful evidence. The experiment motivates bounded retrieval as a means of balancing access to external evidence with focused reasoning.

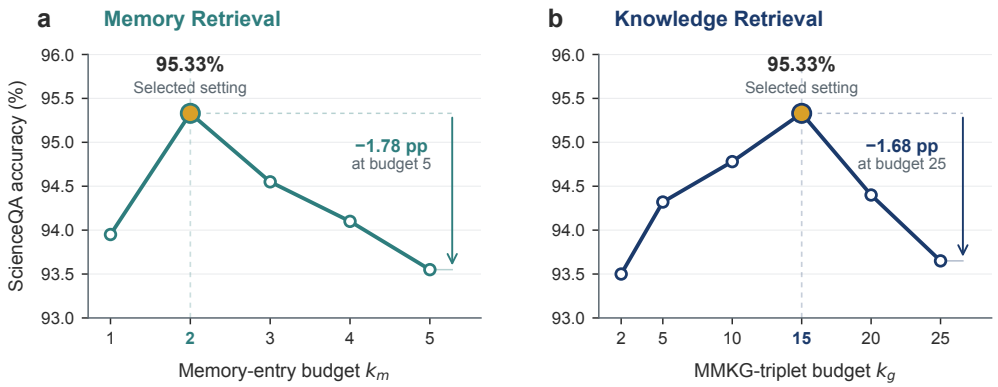


Figure 6: Retrieval-budget sensitivity on ScienceQA-test with Qwen3.5-9B. Gold markers indicate settings selected on the ScienceQA training split ($k_{\mathrm{m}} = 2$ and $k_{\mathrm{g}} = 15$), both attaining the highest observed accuracy of 95.33% in their respective sweeps. Arrows mark the decrease to the largest tested budget in percentage points (pp), showing that larger budgets need not improve performance.

F.5 Evidence Arbitration and Iteration Profile

Figure 7 examines how frequently readiness-gated arbitration invokes further refinement. Across all 4,241 ScienceQA test questions, 97.97% terminate after one outer iteration, with a mean of 1.022 iterations per question. The initial stage can already incorporate memory retrieval, grounded evidence acquisition, and evidence arbitration. The combination of 95.33% overall accuracy and predominantly single-iteration completion highlights an iteration-efficient reasoning pattern, with readiness-guided control dynamically invoking deeper refinement only for cases with unresolved evidence needs. Table 4 provides the complementary evidence for component-level accuracy gains; the iteration profile describes how the complete framework operates under the evaluated conditions.

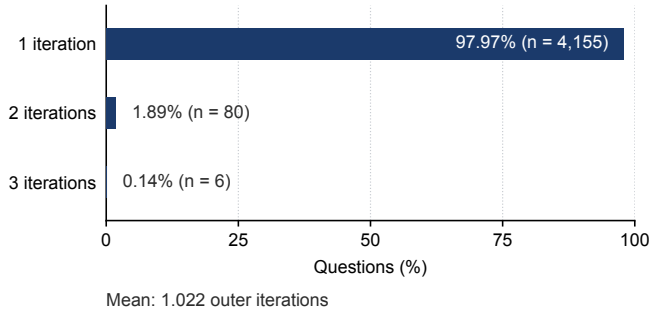


Figure 7: **Selective refinement beyond the initial evidence assessment.** Under readiness-gated arbitration, most questions terminate after one outer iteration, which can already integrate perception, retrieval, and memory-informed reasoning. The small multi-iteration tail shows that further refinement is invoked selectively for unresolved cases. Results cover one Qwen3.5-9B run on the full ScienceQA test split.

G Discussion

Why cooperative verification works. The incremental ablations support complementary roles for evidence arbitration and experience reuse: completing EDAR beyond retrieval alone contributes the largest added gain on both benchmarks, and SHM remains beneficial after its inclusion. EDAR brings grounded candidates and conflicting observations into the fact reasoner's assessment, while SHM makes prior episodes, semantic knowledge, and procedures available across questions. This distinction connects instance-specific evidence assessment with accumulated experience. Both

mechanisms can inform the initial reasoning stage, with further refinement available when the current evidence remains insufficient.

Why scale and the framework are complementary. Within Qwen3.5, the 9B-to-27B transition raised MMMU by 9.02 points, compared with 3.23 on ScienceQA and less than one point on each A-OKVQA format. Qwen3.6-35B further increased direct-answer and MMMU accuracy, although this comparison combines capacity and model-version changes. The stronger MMMU gain may reflect its greater dependence on latent disciplinary knowledge. Verification can protect relevant evidence from contradiction, but the fact reasoner must still possess the concepts required to interpret it.

Broader implications. Closed-loop verification may help other settings where frozen multimodal models require auditable outputs. Typed masks, boxes, and verifier scores support explicit contradiction detection. Similar mechanisms could benefit scientific documents, medical-image triage, and embodied agents, although these extensions require separate evaluation.