EPICON: COLLECTIVE AGENT LEARNING THROUGH CO-EVOLVING MULTIMODAL MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Agents can learn from past executions, but enabling different agents to reuse and build on one another's experience remains challenging. We introduce EpiCon, a shared multimodal memory framework for agent collective learning without updating host model parameters. EpiCon links question-level memory evolution to a persistent experience bank through two independently trained 2B models: a memory controller and a tree self-organizer. The controller jointly refines textual guidance and visual evidence across attempts and selectively includes visual memory. The self-organizer consolidates lessons hierarchically and retrieves experience and rules for new problems. We evaluate EpiCon on eleven benchmarks spanning four multimodal task domains, using two harnesses and multiple backbones. A frozen bank improves other systems even with a single solving attempt. A second harness raises the original system's macro-average score by 2.6 points across eleven benchmarks. Across four host configurations, EpiCon improves macro-average scores by 1.7 to 4.9 points over No Memory and reduces memory-operation time by 67% to 74% relative to backbone-sized memory models.

1 INTRODUCTION

Multi-agent systems (MAS) built on language models combine reasoning, tool use, and collaboration to solve complex tasks (Wu et al., 2023; Fourney et al., 2024). Their executions produce experience about solution procedures, failure modes, and relevant evidence. External memory allows this experience to inform subsequent inference without updating host model parameters (Wang et al., 2024b; Ouyang et al., 2026). As tasks accumulate, a system can both draw on earlier experience and contribute new lessons. This motivates a shared memory that supports continued learning within the same MAS while keeping experience useful across different harnesses and backbones.

Recent work has advanced agent memory through linked notes and hierarchical experience structures (Xu et al., 2026; Zhang et al., 2026b), while shared memory systems enable experience reuse across models and frameworks (Tang et al., 2025; Chang et al., 2026). Multimodal memory preserves visual evidence alongside textual guidance (Zeng et al., 2026b). Lessons may depend on diagram regions or document details, and both guidance and supporting evidence may need revision across attempts. Accumulated lessons also require consolidation for later retrieval and reuse. We study how to connect feedback-driven multimodal refinement with a shared experience bank that different agent systems can reuse and update.

We introduce **EpiCon** (**Epi**sodic **Con**solidation), a shared multimodal memory framework that supports agent collective learning through experience accumulation and reuse. Its dedicated memory harness connects temporary question-level memory with a persistent experience bank. Within a question, a Memory Controller supports textual and visual memory co-evolution, jointly revising actionable guidance and the associated image regions in response to successive attempts and feedback. As the guidance evolves, the controller can revise the visual evidence and adaptively decide whether to include visual memory in the next attempt. Across questions, a Tree Self-Organizer organizes and consolidates lessons in a shared bank maintained independently of the MAS. Accumulated experience can guide later solving within the same MAS and remain useful even if the harness or backbone changes. During bank construction and expansion, different harnesses can both reuse existing experience and contribute new lessons, allowing the evolved bank to support subse-

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.

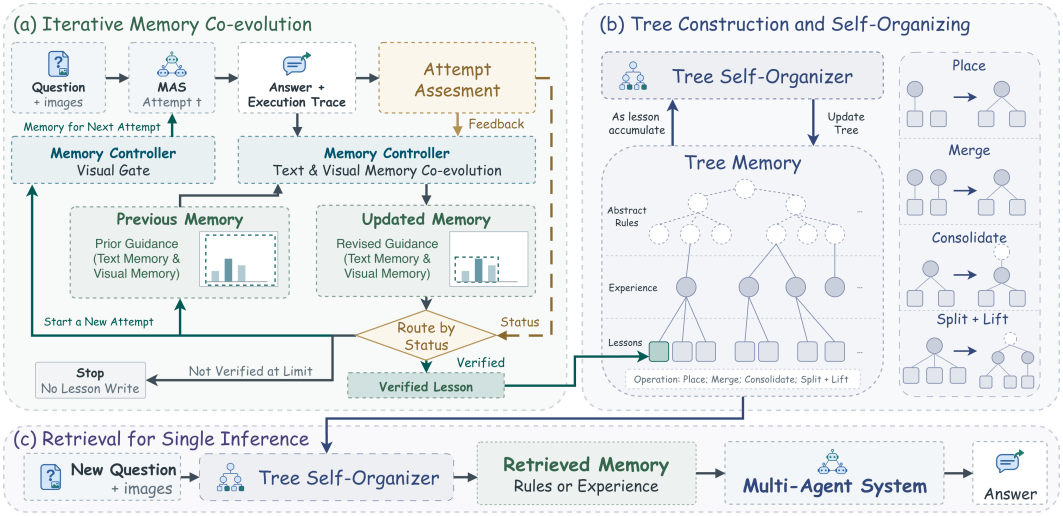

Figure 1: Overview of EpiCon. The memory controller refines question-level memory, while the tree self-organizer maintains accumulated lessons for subsequent retrieval and reuse. The memory harness coordinates these operations with MAS execution. Bank maintenance occurs during construction; the historical bank remains frozen during evaluation.

quent solving by its contributors. We implement memory control and tree organization with two independently trained 2B models.

We evaluate EpiCon on eleven benchmarks across four multimodal task domains. Experiments with two MAS harnesses and multiple backbones show improved task performance and reuse of experience across configurations. An evolved bank that incorporates experience from another harness also improves the original harness's subsequent solving. The trained 2B models reduce memory-operation time relative to backbone-sized memory models, and performance gains extend to Codex with GPT-5.6-Luna. Our contributions are summarized as follows:

- **Collective agent learning through shared multimodal memory.** We introduce **EpiCon**, which lets systems share experience without updating host parameters. Frozen-bank reuse improves performance across harnesses and backbones with a single attempt. Contributions from another harness raise the original system's macro-average score by 2.1–3.2 points.
- **Co-evolution of textual guidance and visual evidence.** A memory controller jointly refines guidance and supporting image regions across attempts and selectively includes visual memory. A tree self-organizer consolidates lessons hierarchically and retrieves experience and rules for new problems.
- **Efficient memory management with compact models.** Both modules are independently trained 2B models. On eleven benchmarks spanning four multimodal domains, the 2B variant improves macro-average scores by 1.7–4.9 points over No Memory across four host configurations and reduces memory-operation time by 67–74% relative to backbone-sized memory models.

2 Related Work

2.1 Multi-Agent Systems

Large language model (LLM) agents are increasingly organized into multi-agent systems (MAS) to divide complex tasks across specialized roles and coordinate complementary capabilities. Existing systems support collaboration through role-based dialogue and structured workflows (Li et al., 2023; Wu et al., 2023; Chen et al., 2024; Hong et al., 2024; Qian et al., 2024), multi-agent debate (Du et al., 2023), and adaptive interaction structures such as optimized agent graphs, orchestration, and team generation (Zhuge et al., 2024; Fourney et al., 2024; Yuan et al., 2025). These approaches make communication and coordination central to MAS design. However, long-horizon and repeated interactions introduce a further requirement: agents must retain useful observations, decisions, failures,

and collaboration patterns beyond the current exchange. Communication determines how information moves among agents, whereas memory determines what remains available over time. Recent work therefore begins to model multi-agent memory explicitly, including agent-specific and cross-trial collaboration histories (Zhang et al., 2026b). Our work treats memory as a persistent substrate through which multimodal experiences from different agents can be organized and reused.

2.2 Agent Memory and Multimodal Experience

Agent memory has evolved from retaining information to managing reusable experience. Early systems store and retrieve interaction histories or episodic records to support persistent behavior (Park et al., 2023; Packer et al., 2023; Zhong et al., 2024). Later work adds extraction, consolidation, hierarchical organization, associative retrieval, dynamic linking, and updating (Chhikara et al., 2025; Xu et al., 2026; Gutiérrez et al., 2024). Other approaches store trajectories, workflows, or learned skills as reusable experience (Zheng et al., 2024; Wang et al., 2023; 2024b; Zeng et al., 2026a). Most language-agent memories remain primarily textual. Multimodal agents preserve perceptual evidence alongside higher-level knowledge for long-horizon reasoning (Li et al., 2024; Long et al., 2026; Zeng et al., 2026b). In MAS, agents may contribute different parts of the same experience, making memory sharing important (Zhang et al., 2026b). EpiCon jointly refines textual guidance and visual evidence across attempts and shares the resulting experience across systems.

2.3 Reflection and Self-Evolving Agents

Reflection and self-improvement study how agents can turn interaction outcomes into better future behavior. Self-refinement and critique methods revise outputs using model-generated or external feedback (Madaan et al., 2023; Shinn et al., 2023; Gou et al., 2024), while reflection- and search-based agents use execution outcomes to guide later attempts (Shinn et al., 2023; Zhou et al., 2023). Experiential-learning methods go further by abstracting trajectories into reusable lessons, workflows, policies, or skills (Zheng et al., 2024; Wang et al., 2023; 2024b; Zhao et al., 2024; Zhang et al., 2024; Zheng et al., 2025). Recent work on self-evolving agents extends this process by integrating successful and failed experience into persistent reasoning memory or broader capability updates (Ouyang et al., 2026; Yan et al., 2026; Yang et al., 2026; Cheng et al., 2026). These works form a progression from retry, to reflection, to experience consolidation. Our work connects multimodal memory evolution within a question with the reuse of accumulated experience across questions. We study these two roles separately, evaluating historical experience reuse with a single solving attempt so that its benefits do not depend on additional attempts.

3 Data Curation

We construct two supervision datasets for memory control and tree organization using Qwen3.8-27B (Qwen Team, 2026b) as the initial teacher and Qwen3.8-Flash-Next (Qwen Team, 2026a) for selective refinement. The initial teacher proposes joint text–visual memory updates, lessons, and tree-operation demonstrations. Qwen3.8-Flash-Next revises unsuccessful updates, improves lesson quality, and corrects tree-operation targets where needed.

Source datasets. We collect experience from tasks in MathNet (Alshammari et al., 2026), MathV360K (Shi et al., 2024), ChartQA (Masry et al., 2022), InfoVQA (Mathew et al., 2022), DocVQA (Mathew et al., 2021), and ChartNet (Kondic et al., 2026). Our training tasks cover three domains: visual mathematics, document understanding, and reasoning over charts and infographics. MAS executions on these tasks provide the records we use to construct memory updates, lessons, and tree-operation supervision.

Memory controller. We collect MAS trajectories using CAMEL (Li et al., 2023), Codex (OpenAI, 2026a), and DeepSeek-Harness (DeepSeek-AI, 2026). Given the question, images, previous memory, latest attempt, and correctness feedback, the teachers generate and refine joint text–visual memory updates. We replay the previous and proposed memories with the same MAS host, retaining Repair transformations that turn failed attempts into successful ones and Compress transformations that preserve success while shortening text without increasing crop area. Reference answers are used to assess attempts during this offline screening. Accepted joint updates provide approximately 25K supervision examples.

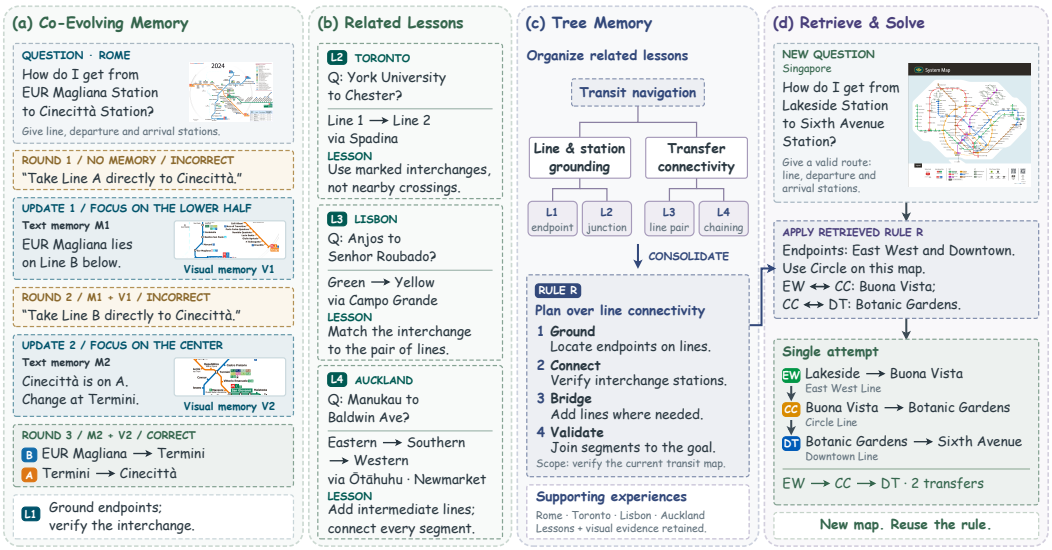

Figure 2: Illustrative example of EpiCon. (a) Textual and visual memory co-evolve across attempts. (b) and (c) Related lessons are organized and consolidated into a reusable routing rule. (d) Retrieved guidance supports a single solving attempt on a new map. Trajectories and tree operations are reconstructed for illustration.

Tree self-organizer. Lessons derived from the collected controller memories seed small memory banks, with historical lessons separated from retrieval queries. The teachers generate and selectively refine demonstrations of placement, merging, splitting, consolidation, and retrieval. We retain approximately 6.5K operation demonstrations that pass schema, node-reference, partition, and provenance checks, including valid empty retrievals when no candidate applies. These checks establish structural validity; individual tree operations are not validated through downstream replay. Appendix A provides construction and filtering details.

4 METHODOLOGY

EpiCon combines a dedicated memory harness with two independently trained 2B models: a Memory Controller C_θ and a Tree Self-Organizer S_ϕ (Figure 1). The controller updates textual and visual memory within a question, while the organizer maintains a shared experience bank for reuse across questions and agent systems. The harness coordinates memory updates, retrieval, visual cropping, and storage. Both models are trained on the supervision described in Section 3; the host MAS retains its own orchestration and backbone, whose parameters remain unchanged.

4.1 SHARED MULTIMODAL MEMORY

EpiCon maintains a persistent experience bank B independently of the MAS that uses it. A system can retrieve guidance from this bank and contribute lessons from its own task executions. These lessons support subsequent tasks within the same MAS and can also be reused across different harnesses and backbones. Collective learning therefore takes place through the accumulation and reuse of shared experience, without updating host model parameters.

For a question x_i with its images and public task information, the organizer retrieves experience R_i once from B. The MAS uses the retrieved experience to produce an answer $\hat{y}_{i,t}$ together with an execution trace $\tau_{i,t}$. When question-level evolution is enabled, the controller maintains a separate memory $W_{i,t}$, initially empty, and revises it using the question, images, existing memory, answer, and execution trace. Our evaluations of historical experience reuse disable this loop and use a single solving attempt.

During bank construction or expansion, accepted lessons and their revisions are incorporated into B, allowing experience to accumulate across tasks. Let $\mathcal{U}_k$ denote the batch of accepted lessons and

Table 1: Main results across two MAS harnesses, two backbones, and eleven benchmarks. All methods make a single solving attempt per question, without question-level memory updates. Both EpiCon variants use MC + Tree. Parentheses identify the memory models: ours 2B uses two independently trained 2B models; named backbones operate zero-shot. Time and token usage separate MAS execution from memory operations.

MAS Harness	MAS Backbone	Memory System	Document Understanding			Visual-to-Code			Vision-Grounded Math		General VL Reasoning			Time (×)		Tokens (×)	
			DocVQA2026	MP-DocVQA	ParseBench	VisionCode	Omni-LEC	ChartMimic	MATH-Vision	WeMath	WorldSense	ReasonMap	RoboVision	MAS	Memory	MAS	Memory
Codex	Qwen3.8-27B	No Memory	15.7	79.5	57.1	53.0	66.2	66.4	20.4	81.0	49.2	40.1	14.2	1.00	0.00	1.00	0.00
		Mem0 (Qwen3.8-27B)	1.4	81.6	61.9	61.3	70.6	78.0	24.4	80.6	48.6	39.6	12.2	1.33	0.31	1.23	0.11
		Cognee (Qwen3.8-27B)	14.3	73.7	60.1	61.5	69.9	78.1	44.2	78.2	51.2	35.4	11.5	1.16	1.66	1.11	0.75
		A-Mem (Qwen3.8-27B)	12.9	80.2	66.3	60.0	72.3	78.6	25.4	81.8	45.4	40.6	13.5	1.23	0.24	1.27	0.81
		Agent-KB (Qwen3.8-27B)	11.4	78.6	59.5	25.8	67.4	69.8	25.2	80.8	45.0	24.1	15.6	1.54	0.31	1.12	0.14
		EpiCon (ours 2B)	18.6	66.1	67.3	59.9	71.4	75.2	45.2	84.2	52.4	41.5	15.3	1.16	0.30	1.04	0.67
		EpiCon (Qwen3.8-27B)	**22.9**	**83.5**	**68.5**	**61.7**	**73.0**	**79.3**	**48.6**	**85.4**	**53.8**	**43.4**	**16.3**	1.15	0.92	1.03	0.64
	Gemma4-31B	No Memory	5.7	46.2	43.3	51.7	65.9	65.6	25.8	85.0	45.4	31.6	17.7	1.00	0.00	1.00	0.00
		Mem0 (Gemma4-31B)	7.1	54.0	33.4	51.2	65.1	62.2	29.2	86.2	44.4	27.8	12.2	1.40	0.53	1.24	0.12
		Cognee (Gemma4-31B)	4.3	49.9	27.6	54.6	49.0	63.7	37.4	72.4	42.2	27.4	12.8	0.92	7.23	1.07	1.02
		A-Mem (Gemma4-31B)	**8.6**	55.5	37.0	54.9	**68.6**	63.1	29.8	88.0	44.6	31.6	13.9	1.23	0.39	1.31	0.97
		Agent-KB (Gemma4-31B)	5.7	51.4	27.4	52.8	62.1	45.9	26.2	87.2	46.8	32.5	16.3	1.41	0.31	1.04	0.13
		EpiCon (ours 2B)	**8.6**	58.8	44.2	54.3	67.5	**66.7**	38.8	87.8	53.2	32.1	18.1	1.33	0.47	1.01	0.74
		EpiCon (Gemma4-31B)	**8.6**	**59.6**	**45.8**	**57.1**	67.8	66.4	**57.6**	**88.4**	**54.2**	**36.3**	**19.1**	1.20	1.56	1.02	0.67
Deepseek Harness	Qwen3.8-27B	No Memory	14.3	79.5	52.4	53.7	61.4	65.3	42.2	88.8	54.0	22.6	20.1	1.00	0.00	1.00	0.00
		Mem0 (Qwen3.8-27B)	11.4	80.5	46.1	54.9	61.0	**68.8**	38.8	90.0	52.2	21.2	17.0	1.10	0.15	1.87	0.41
		Cognee (Qwen3.8-27B)	15.7	76.6	50.9	54.0	60.6	68.3	53.6	91.2	50.2	19.3	18.4	0.95	0.85	1.33	2.70
		A-Mem (Qwen3.8-27B)	17.1	80.6	55.4	53.7	63.4	65.5	43.6	89.6	52.2	22.2	17.4	1.09	0.12	2.01	2.98
		Agent-KB (Qwen3.8-27B)	**20.0**	79.5	50.5	52.5	62.6	63.8	42.6	90.4	52.6	13.7	20.5	1.13	0.15	1.51	0.64
		EpiCon (ours 2B)	18.6	81.8	53.6	56.9	**63.5**	65.2	56.6	**92.2**	57.4	24.5	17.4	1.08	0.15	1.06	1.92
		EpiCon (Qwen3.8-27B)	**20.0**	**82.9**	**57.0**	**61.2**	62.8	65.4	**58.2**	91.8	**58.2**	**27.4**	**23.3**	1.07	0.57	1.10	2.88
	Gemma4-31B	No Memory	10.0	2.7	35.6	51.4	55.4	32.8	16.0	91.0	49.6	27.4	11.5	1.00	0.00	1.00	0.00
		Mem0 (Gemma4-31B)	7.1	9.1	35.7	51.6	51.6	32.6	11.4	88.6	49.2	26.4	12.5	0.93	0.20	1.80	0.38
		Cognee (Gemma4-31B)	5.7	0.5	37.7	49.7	44.8	29.9	12.6	89.4	49.8	**30.7**	12.8	0.82	2.23	1.22	3.06
		A-Mem (Gemma4-31B)	7.1	**10.8**	38.8	51.0	52.4	33.0	**20.2**	89.6	52.2	24.1	12.8	0.87	0.12	1.99	3.02
		Agent-KB (Gemma4-31B)	7.1	1.8	36.1	50.0	51.5	29.5	13.0	89.2	47.8	25.5	15.6	0.86	0.18	1.38	0.60
		EpiCon (ours 2B)	5.7	2.0	41.0	52.9	**57.5**	**34.3**	18.8	91.2	**54.0**	28.8	16.3	0.86	0.14	1.08	1.70
		EpiCon (Gemma4-31B)	**11.4**	0.7	**46.3**	**54.3**	56.8	32.9	18.4	**92.2**	53.2	29.2	**17.4**	0.86	0.53	1.09	2.25

revisions processed at maintenance step k:

$$B^{(k+1)} = S_\phi(B^{(k)}, \mathcal{U}_k). \tag{1}$$

The index k counts maintenance steps, not individual lessons or questions, and a step can process multiple lessons together. The update represents the organizer's proposals after validation and execution by the memory harness. During evaluation, the shared bank is frozen and updates remain local to the current question.

4.2 Question-Level Multimodal Memory Evolution

Textual and visual memory co-evolution. Question-level memory is represented as $W_{i,t} = (M_{i,t}^{\text{text}}, M_{i,t}^{\text{vis}})$. Text records actionable guidance, applicability conditions, and cautions; visual memory points to supporting regions in source images. When another attempt is available, the controller updates memory using the previous state, current question and images, latest answer, and execution trace:

$$W_{i,t} = C_\theta^{\text{update}}(W_{i,t-1}, x_i, \hat{y}_{i,t}, \tau_{i,t}). \tag{2}$$

The equation denotes the accepted update after harness validation. The controller jointly proposes revised text, a source image, and a region. It can retain, replace, or remove visual evidence as the guidance evolves. The harness validates the proposal and generates the requested crop; invalid proposals leave memory unchanged. The accepted text and visual evidence guide the next attempt.

Adaptive visual memory injection. The controller decides whether visual memory should be included in the next input through $g_{i,t} \in \{0, 1\}$. For retrieved experience, the decision is based on the current question and descriptions of the retrieved memory. For question-level memory, the joint update supplies the decision together with the revised text and region. Let $T_{i,t}$ and $V_{i,t}$ denote the text and available visual memory for attempt t, drawn from retrieved experience or the latest question-level state. The memory input is

$$c_{i,t} = \begin{cases} (T_{i,t}, \varnothing), & g_{i,t} = 0, \\ (T_{i,t}, V_{i,t}), & g_{i,t} = 1. \end{cases} \tag{3}$$

Here $V_{i,t}$ is bounded by the memory-image budget. The gate selects input content without deleting stored visual memory. The current question's original images remain available in both cases.

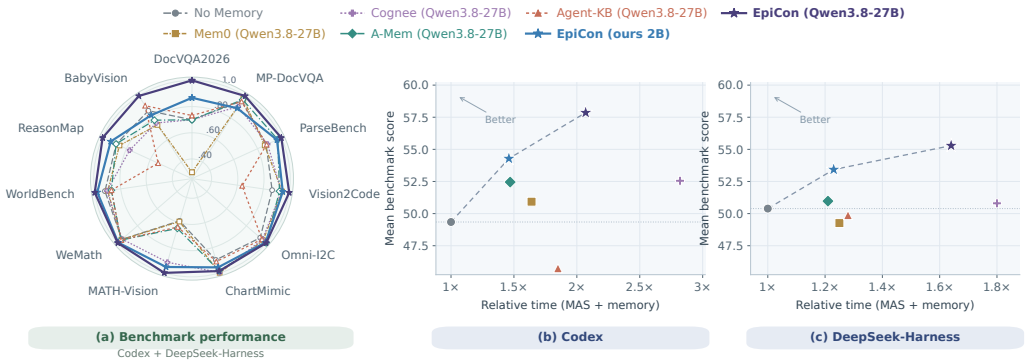

Figure 3: Benchmark profiles and performance–time trade-offs with Qwen3.8-27B. (a) Scores are averaged across the two harnesses and normalized by the best method mean per benchmark; the radial axis starts at 0.25. (b) and (c) show scores averaged equally over eleven benchmarks; time sums normalized MAS and memory costs. Dashed lines connect No Memory and the two EpiCon variants as visual guides.

4.3 Shared Experience Consolidation and Reuse

Organizing accumulated lessons. The shared bank uses a tree to organize lessons with their guidance, source references, and visual evidence. Leaves store question-specific lessons, while internal nodes group and summarize related experience. During maintenance, the Tree Self-Organizer can propose Place, Merge, Split + Lift, and Consolidation operations to assign categories, combine related groups, separate broad groups, and summarize shared procedures and conditions. The harness validates and executes these proposals and can enforce capacity-based partitions.

Abstracting reusable rules. Consolidation captures common guidance and conflicts across lessons. The harness determines abstraction levels and rule eligibility from support, failures, conflicts, and evidence strength, while the organizer generates summaries. Internal nodes can retain qualified summaries or become reusable rules when the evidence permits. This organization allows later tasks to access specific experience and guidance consolidated from multiple lessons.

Retrieving experience for subsequent tasks. Retrieval first recalls candidates and then selects relevant experience:

$$R_i = S_\phi^{\text{retrieve}}(x_i, \text{Recall}(x_i, B)) \,. \tag{4}$$

Here Recall uses a frozen text encoder to retrieve candidates from node indices describing their topic, scope, and question anchor. The organizer examines the current question and candidate memories, including textual guidance and associated visual evidence. It selects applicable rules, a concrete experience, or nothing when no candidate applies. The selected content is available to the receiving MAS regardless of which harness produced it, with visual delivery controlled by C_θ. Retrieval uses existing bank content; further construction or expansion can add lessons and revise its organization for subsequent use. Figure 2 illustrates how question-level memory co-evolution connects to experience consolidation and reuse across transit maps.

5 Experiments

5.1 Experimental Setup

Baselines and model configurations. Our main experiments use two multi-agent system (MAS) harnesses, Codex (OpenAI, 2026a) and DeepSeek-Harness (DeepSeek-AI, 2026), each paired with Qwen3.8-27B (Qwen Team, 2026b) and Gemma4-31B (Team, 2026). We additionally evaluate Codex with GPT-5.6-Luna (OpenAI, 2026b). We compare EpiCon with No Memory, Mem0 (Chhikara et al., 2025), Cognee (Markovic et al., 2025), A-Mem (Xu et al., 2026), and Agent-KB (Tang et al., 2025). External memory systems use the corresponding MAS backbone zero-shot for memory operations. Backbone-sized EpiCon uses the same backbone for memory control and tree organization; the 2B variant uses two independently trained 2B models.

Evaluation suite. We evaluate 4,538 questions from 11 benchmarks spanning document understanding, visual-to-code generation, vision-grounded mathematics, and general visual-language rea-

Table 2: Cross-backbone and cross-harness experience transfer from a bank built by Codex with Qwen3.8-27B. The evaluated MAS performs one solve; EpiCon (ours 2B) reuses the frozen bank without question-level updates.

MAS Harness	MAS Backbone	Memory System	Document Understanding			Visual-to-Code			Vision-Grounded Math		General VL Reasoning		
			DocVQA2026	MP-DocVQA	ParseBench	Vision2Code	Omni-I2C	ChartMimic	MATH-Vision	WeMath	WorldBench	ReasonMap	BabyVision
Codex	Gemma4-31B	No Memory	**5.7**	46.2	43.3	51.7	65.9	65.6	25.8	**85.0**	45.4	31.6	17.7
		EpiCon (ours 2B)	**5.7**	**60.4**	**45.3**	**53.8**	**67.8**	**68.2**	**33.0**	**85.0**	**47.8**	**34.4**	**18.1**
DeepSeek Harness	Qwen3.8-27B	No Memory	14.3	**79.5**	52.4	53.7	61.4	65.3	42.2	88.8	**54.0**	22.6	20.1
		EpiCon (ours 2B)	**18.6**	79.1	**65.7**	**59.2**	**63.3**	**68.7**	**56.0**	**90.6**	52.2	**26.4**	**20.5**

Table 3: Cross-harness memory evolution with EpiCon (ours 2B). Each evaluated harness builds the original bank; the other harness uses and evolves it. The two stages each use 1,000 disjoint construction questions. Both bank conditions use a single solving attempt.

MAS Harness	MAS Backbone	Memory Bank	Document Understanding			Visual-to-Code			Vision-Grounded Math		General VL Reasoning		
			DocVQA2026	MP-DocVQA	ParseBench	Vision2Code	Omni-I2C	ChartMimic	MATH-Vision	WeMath	WorldBench	ReasonMap	BabyVision
Codex	Qwen3.8-27B	Original bank	**18.6**	66.1	**63.4**	59.9	**71.4**	75.2	45.2	84.2	52.4	38.3	14.3
		Evolved bank	**18.6**	**80.3**	58.9	**61.3**	70.9	**76.7**	**45.4**	**88.4**	**55.2**	**40.1**	**16.8**
DeepSeek Harness	Qwen3.8-27B	Original bank	18.6	81.8	52.3	56.9	63.5	65.2	**56.6**	**92.2**	57.4	23.5	19.3
		Evolved bank	**20.0**	**82.7**	**65.9**	**58.7**	**69.2**	**72.1**	53.4	91.2	**57.6**	**29.6**	**22.3**

soning: DocVQA2026 (Llabrés et al., 2027), MP-DocVQA (Tito et al., 2022), ParseBench (Zhang et al., 2026a), Vision2Code (Periasami et al., 2026), Omni-I2C (Zhou et al., 2026), ChartMimic (Yang et al., 2025), MATH-Vision (Wang et al., 2024a), WeMath (Qiao et al., 2025), WorldBench (Yin et al., 2026), ReasonMap (Feng et al., 2026), and BabyVision (Chen et al., 2026).

Evaluation protocol. Across all experiments, No Memory makes a single solving attempt per question. For the main comparison, each memory-based method builds its own bank using a shared set of 1,000 construction questions from the same benchmarks, disjoint from the evaluation questions under our project-defined splits. ReasonMap is split by map, with no map shared between construction and evaluation. All evaluations with historical memory retrieval use a single solving attempt per question, with question-level memory updates disabled. In Table 1, this single-attempt protocol applies to every method. Each attempt follows the harness's standard internal workflow. Model parameters and historical banks remain frozen during evaluation. Reference answers are used only for offline scoring. We report scores, time, and token usage, with MAS and memory costs separated. Time and token usage are normalized to the corresponding No Memory MAS costs. All experiments are run using NVIDIA H100 GPUs; GPT-5.6-Luna is accessed through its API.

5.2 Main Results

With a single solving attempt per question, backbone-sized EpiCon achieves the highest eleven-benchmark macro-average in all four host configurations, exceeding the strongest external memory baseline in each by 1.9 to 5.9 points (Table 1). Macro-averages give equal weight to each benchmark. For Codex with Qwen3.8-27B, this variant improves ParseBench from 57.1 to 68.5 and MATH-Vision from 20.4 to 48.6 over No Memory. Averaged over the two Qwen3.8-27B host configurations, it also scores highest on ten of eleven benchmarks (Figure 3(a)).

Replacing backbone-sized memory models with our two trained 2B models reduces memory-operation time by approximately 67% to 74% and total (MAS + memory) time by 25% to 35%, with macro-average scores lower by 0.9 to 3.6 points. The 2B variant still improves macro-average scores over No Memory by 1.7 to 4.9 points across the four host configurations. In Figure 3(b) and (c), both EpiCon configurations lie on the empirical Pareto front among the compared methods for each harness with Qwen3.8-27B.

Our two 2B memory models are trained on supervision from three domains: visual mathematics, document understanding, and reasoning over charts and infographics (Section 3). They also support gains on general visual-language reasoning: the 2B variant improves WorldBench in all four configurations ($+3.2$ to $+7.8$ points), while changes on ReasonMap ($+0.5$ to $+1.9$) and BabyVision (-2.7 to $+4.8$) are smaller and less consistent. These results suggest that the trained memory models can manage experience from additional task types without further parameter updates.

Table 4: Question-level memory control with no memory bank. No Memory makes a single solving attempt; MC settings allow up to five attempts. **MC (Qwen3.8-27B)** uses Qwen3.8-27B as a zero-shot memory controller and **MC (ours 2B)** uses our trained 2B controller. We report task performance on four benchmarks. Time and token usage are normalized separately within each MAS harness using the corresponding No Memory MAS cost as the $1\times$ baseline; the same baseline normalizes memory costs.

MAS	MAS Backbone	Memory Setting	Task Performance				Time (×)		Tokens (×)	
			Document ParseBench	Visual-to-Code Vision2Code	Vision Math MATH-Vision	General VL BabyVision	MAS	Memory	MAS	Memory
Codex	Qwen3.8-27B	No Memory	57.1	53.0	20.4	14.2	1.00	0.00	1.00	0.00
		MC (ours 2B)	59.8	55.9	41.2	**17.7**	2.16	0.23	2.40	0.96
		MC (Qwen3.8-27B)	**63.1**	**56.6**	**49.6**	13.9	2.57	0.23	3.07	0.95
DeepSeek Harness	Qwen3.8-27B	No Memory	52.4	53.7	42.2	20.1	1.00	0.00	1.00	0.00
		MC (ours 2B)	55.1	57.8	51.4	23.3	2.10	0.13	2.44	3.23
		MC (Qwen3.8-27B)	**64.5**	**65.8**	**61.6**	**35.1**	2.47	0.16	2.65	3.25

5.3 Experience Transfer and Memory Expansion

We evaluate experience reuse across harnesses and backbones, followed by continued accumulation in a shared bank. All comparisons use EpiCon with fixed Memory Controller (MC) and Tree Self-Organizer (Tree) models.

Cross-backbone and cross-harness transfer. A bank built by Codex with Qwen3.8-27B is reused by Codex with Gemma4-31B and DeepSeek-Harness with Qwen3.8-27B, changing only the backbone or the harness, respectively. Each evaluated MAS performs one solve with or without the frozen bank, disabling question-level updates to assess historical experience reuse. Table 2 compares performance across all eleven benchmarks, with macro-average improvements of 3.2 and 4.2 points for backbone and harness transfer, respectively. For example, backbone transfer improves MATH-Vision from 25.8 to 33.0, while harness transfer improves ParseBench from 52.4 to 65.7. These gains show that a system can benefit from existing shared experience without first constructing its own bank, even when its harness or backbone differs from that of the original contributor.

Cross-harness memory evolution. We test whether experience contributed by another harness can benefit the harness that built the bank. One harness builds the original bank on 1,000 construction questions; the other uses and evolves it on a separate set of 1,000 questions. We evaluate both directions between Codex and DeepSeek-Harness, with Qwen3.8-27B as the backbone throughout. In each direction, the harness that built the original bank is evaluated with the original and evolved banks on the same 4,388 questions across eleven benchmarks, disjoint from both construction sets. Both conditions use one retrieval and a single solving attempt, without question-level memory updates. Table 3 compares the original and evolved banks. Macro-average scores rise from 53.5 to 55.7 for Codex and from 53.4 to 56.6 for DeepSeek-Harness, gains of 2.1 and 3.2 points. The evolved banks improve eight and nine of eleven benchmark scores, respectively. The comparison measures the overall effect of bank evolution and expansion; construction splits are provided in Appendix C.

The benefits of cross-harness evolution depend on the task and contributing system. On ParseBench, where Codex is stronger in Table 1, its score drops from 63.4 to 58.9 after DeepSeek-Harness evolves its bank, while DeepSeek-Harness improves from 52.3 to 65.9 after Codex evolves its bank. On MATH-Vision, Codex changes only slightly from 45.2 to 45.4, while DeepSeek-Harness declines from 56.6 to 53.4. These task-dependent patterns suggest that experience from different systems can provide complementary guidance, although individual benchmarks do not improve uniformly. Taken together, Tables 2 and 3 support collective learning through shared memory: a system can benefit from experience accumulated by another system, then contribute additional experience that improves subsequent solving by the original contributor.

5.4 Memory Ablations

Question-level memory control. With historical retrieval disabled, Table 4 compares No Memory, MC (ours 2B), and MC (Qwen3.8-27B). No Memory makes a single solving attempt, while MC revises memory across attempts using the question, images, existing memory, answer, and execution trace. Refinement stops after at most five attempts or earlier when the system judges the answer ready to return. With the trained 2B controller, refinement improves all four scores under both harnesses, including Vision2Code from 53.7 to 57.8 with DeepSeek-Harness. The larger controller

Table 5: Multimodal memory ablation with trained MC 2B and no historical bank. Memory-enabled settings share a budget of up to five attempts; the row with three dashes in the memory configuration columns denotes No Memory with a single solving attempt. Visual memory can be frozen or evolving, and is injected either on every attempt (Always) or adaptively (Adaptive). Original task images remain available in every setting.

MAS	MAS Backbone	Memory Configuration			Task Performance				Time (×)		Tokens (×)	
		Text Memory	Visual Memory	Visual Injection	Document ParseBench	Visual-to-Code Vision2Code	Vision Math MATH-Vision	General VL BabyVision	MAS	Memory	MAS	Memory
Codex	Qwen3.8-27B	–	–	–	57.1	53.0	20.4	14.2	1.00	0.00	1.00	0.00
		✓	–	–	58.7	55.7	27.2	16.0	1.43	0.28	1.23	0.61
		✓	Frozen	Always	55.2	55.0	40.6	16.0	2.82	0.43	3.00	0.94
		✓	Frozen	Adaptive	57.1	**55.9**	41.0	16.3	2.02	0.25	2.42	0.94
		✓	Evolving	Always	55.3	55.2	41.0	16.3	2.89	0.44	3.02	0.93
		✓	Evolving	Adaptive	**59.8**	**55.9**	**41.2**	**17.7**	2.16	0.23	2.40	0.96
DeepSeek Harness	Qwen3.8-27B	–	–	–	52.4	53.7	42.2	20.1	1.00	0.00	1.00	0.00
		✓	–	–	51.8	55.1	47.4	21.2	1.33	0.13	1.25	2.37
		✓	Frozen	Always	51.2	57.5	50.2	20.5	2.52	0.21	3.13	3.75
		✓	Frozen	Adaptive	52.8	57.3	50.8	21.2	1.95	0.13	2.41	3.58
		✓	Evolving	Always	52.1	57.0	**51.8**	21.2	2.53	0.21	3.15	3.72
		✓	Evolving	Adaptive	**55.1**	**57.8**	51.4	**23.3**	2.10	0.13	2.44	3.23

Table 6: Memory organization with EpiCon (ours 2B). Flat and tree memory use the same source questions to create lessons.

Memory	ParseBench	Vision2Code	MATH-Vision	BabyVision
Flat	52.2	57.5	43.6	13.2
Tree	**67.3**	**59.9**	**45.2**	**15.3**

Table 7: Memory effectiveness on Codex with GPT-5.6-Luna using EpiCon (ours 2B). Both settings make a single solving attempt.

Memory	ParseBench	Vision2Code	MATH-Vision	BabyVision
No Memory	63.8	65.8	51.2	11.1
EpiCon	**66.5**	**67.5**	**57.2**	**14.6**

scores higher in seven of eight comparisons, while the 2B controller requires less MAS execution time and no more memory-operation time.

Visual memory and adaptive injection. Table 5 fixes MC (ours 2B), disables historical retrieval, and uses the same budget of up to five attempts across memory-enabled settings. No Memory makes a single solving attempt; original task images remain available throughout. With adaptive injection fixed, evolving visual memory improves seven scores over frozen memory and ties the remaining score, supporting the benefit of memory co-evolution beyond selective injection. With always-on injection, frozen visual memory improves Codex's MATH-Vision score from 27.2 to 40.6 over text-only memory. Adaptive injection improves seven of eight scores with either frozen or evolving visual memory. For evolving memory, it also reduces MAS execution time by approximately 25% with Codex and 17% with DeepSeek-Harness, with lower MAS token usage under both harnesses.

Experience organization. Table 6 compares flat and tree memory using lessons derived from the same source questions with the trained MC 2B. The flat bank stores lessons without hierarchical organization or consolidation, while the tree bank organizes and consolidates them for retrieval. Both configurations use the same harness, backbone, attempt budget, and memory-context budget. Tree memory improves all scores, including ParseBench from 52.2 to 67.3 and BabyVision from 13.2 to 15.3. These gains support organizing and consolidating experience for subsequent reuse.

5.5 Generalization to a Stronger Solver

We evaluate Codex with GPT-5.6-Luna to test whether EpiCon remains useful with a stronger API-based solver beyond the open-source backbones. The 2B memory components and historical bank remain fixed, without target-specific retraining. Both No Memory and EpiCon make a single solving attempt; EpiCon retrieves experience from the frozen bank. Table 7 shows improvements on all four benchmarks, including ParseBench from 63.8 to 66.5.

6 Conclusion

We presented EpiCon, which connects question-level textual and visual memory co-evolution with the accumulation and reuse of shared experience. Our experiments show that experience can remain useful beyond the system that generated it: another harness can benefit from the existing bank and contribute new experience that improves subsequent solving by the original contributor. These findings support agent collective learning through shared external multimodal memory, allowing systems to build on one another's experience while host model parameters remain fixed.

REPRODUCIBILITY STATEMENT

We will release all source code, model checkpoints, and configuration files necessary to reproduce the results reported in this paper.

AI USE STATEMENT

Generative Al was used to assist in outlining and language refinement for this work. All final arguments, analysis, factual verification, and editorial decisions remain solely the responsibility of the author.

REFERENCES

Shaden Alshammari, Kevin Wen, Abrar Zainal, Mark Hamilton, Navid Safaei, Albarakati Albarakati, William Freeman, and Antonio Torralba. Mathnet: A global multimodal benchmark for mathematical reasoning and retrieval. In *International Conference on Learning Representations*, volume 2026, pp. 12477–12507, 2026.

Yurui Chang, Yiran Wu, Qingyun Wu, and Lu Lin. Memcollab: Cross-agent memory collaboration via contrastive trajectory distillation. *arXiv e-prints*, pp. arXiv–2603, 2026.

Liang Chen, Weichu Xie, Yiyan Liang, Hongfeng He, Hans Zhao, Zhibo Yang, Zhiqi Huang, Haoning Wu, Haoyu Lu, Yiping Bao, et al. Babyvision: Visual reasoning beyond language. *arXiv preprint arXiv:2601.06521*, 2026.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, et al. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *International Conference on Learning Representations*, volume 2024, pp. 20094–20136, 2024.

Zihao Cheng, Zeming Liu, Yingyu Shan, Xinyi Wang, Xiangrong Zhu, Yunpu Ma, Hongru Wang, Yuhang Guo, Wei Lin, and Yunhong Wang. Mem2evolve: Towards self-evolving agents via co-evolutionary capability expansion and experience distillation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20784–20831, 2026.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

DeepSeek-AI. Deepseek harness: Everything is a plugin. `https://github.com/deepseek-ai/deepseek-harness`, 2026.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

Sicheng Feng, Song Wang, Shuyi Ouyang, Lingdong Kong, Zikai Song, Jianke Zhu, Huan Wang, and Xinchao Wang. Reasonmap: Towards fine-grained visual reasoning from transit maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 41077–41088, 2026.

Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, et al. Magentic-one: A generalist multi-agent system for solving complex tasks. *arXiv preprint arXiv:2411.04468*, 2024.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yujiu Yang, Nan Duan, Weizhu Chen, et al. Critic: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*, volume 2024, pp. 57734–57811, 2024.

Bernal J Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in neural information processing systems*, 37:59532–59569, 2024.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Steven Yau, Zijuan Lin, Liyang Zhou, et al. Metagpt: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations*, volume 2024, pp. 23247–23275, 2024.

Jovana Kondic, Pengyuan Li, Dhiraj Joshi, Isaac Sanchez, Ben Wiesel, Shafiq Abedin, Amit Alfassy, Eli Schwartz, Daniel Caraballo, Yagmur Gizem Cinar, et al. Chartnet: A million-scale, high-quality multimodal dataset for robust chart understanding. *arXiv preprint arXiv:2603.27064*, 2026.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in neural information processing systems*, 36:51991–52008, 2023.

Zaijing Li, Yuquan Xie, Rui Shao, Gongwei Chen, Dongmei Jiang, and Liqiang Nie. Optimus-1: Hybrid multimodal memory empowered agents excel in long-horizon tasks. *Advances in neural information processing systems*, 37:49881–49913, 2024.

Artemis Llabrés, Marc Serra Ortega, Tomàs Ockier, Samuel Ortega Cuadra, Amritpal Singh, Christos Georgakilas, Andrey Barsky, Ernest Valveny, and Dimosthenis Karatzas. Icdar2026 competition on multimodal reasoning over documents in multiple domains. In Gernot A. Fink, Alicia Fornés, Koichi Kise, and Daniel Lopresti (eds.), *Document Analysis and Recognition – ICDAR 2026*, pp. 336–352, Cham, 2027. Springer Nature Switzerland. ISBN 978-3-032-36042-7.

Lin Long, Yichen He, Wentao Ye, Yiyuan Pan, Yuan Lin, Hang Li, Junbo Zhao, and Wei Li. Seeing, listening, remembering, and reasoning: A multimodal agent with long-term memory. In *International Conference on Learning Representations*, volume 2026, pp. 146197–146246, 2026.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

Vasilije Markovic, Lazar Obradovic, Laszlo Hajdu, and Jovan Pavlovic. Optimizing the interface between knowledge graphs and llms for complex reasoning. *arXiv preprint arXiv:2505.24478*, 2025.

Ahmed Masry, Jia Qing Tan, Shafiq Joty, Enamul Hoque, et al. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pp. 2263–2279, 2022.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 2199–2208. IEEE, 2021.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2582–2591. IEEE, 2022.

OpenAI. Codex CLI. `https://learn.chatgpt.com/docs/codex/cli`, 2026a. Official documentation. Accessed: 2026-09-20.

OpenAI. GPT-5.6 Luna. `https://developers.openai.com/api/docs/models/gpt-5.6-luna`, 2026b. Official model documentation. Accessed: 2026-09-20.

Siru Ouyang, Jun Yan, I Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long Le, Samira Daruki, Xiangru Tang, et al. Reasoningbank: Scaling agent self-evolving with reasoning memory. In *International Conference on Learning Representations*, volume 2026, pp. 94327–94354, 2026.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Joon Sung Park, Joseph O'Brien, Carrie Jun 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*, pp. 1–22, 2023.

Ajay Vikram Periasami, Junlin Wang, and Bhuwan Dhingra. Vision2code: A multi-domain benchmark for evaluating image-to-code generation. *arXiv preprint arXiv:2605.11307*, 2026.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, et al. Chatdev: Communicative agents for software development. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 15174–15186, 2024.

Runqi Qiao, Qiuna Tan, Guanting Dong, MinhuiWu MinhuiWu, Chong Sun, Xiaoshuai Song, Jiapeng Wang, Zhuoma Gongque, Shanglin Lei, Yifan Zhang, et al. We-math: Does your large multimodal model achieve human-like mathematical reasoning? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20023–20070, 2025.

Qwen Team. Qwen3.8-Flash-Next: A new architecture, towards ultimate cost-efficiency, August 2026a. URL https://qwen.ai/blog?id=qwen3.8-flash-next.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026b. URL https://qwen.ai/blog?id=qwen3.8.

Wenhao Shi, Zhiqiang Hu, Yi Bin, Junhua Liu, Yang Yang, See Kiong Ng, Lidong Bing, and Roy Ka-Wei Lee. Math-llava: Bootstrapping mathematical reasoning for multimodal large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 4663–4680, 2024.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Xiangru Tang, Tianrui Qin, Tianhao Peng, Ziyang Zhou, Daniel Shao, Tingting Du, Xinming Wei, Peng Xia, Fang Wu, He Zhu, et al. Agent kb: Leveraging cross-domain experience for agentic problem solving. *arXiv preprint arXiv:2507.06229*, 2025.

Gemma Team. Gemma 4 technical report, 2026. URL https://arxiv.org/abs/2607.02770.

Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. Hierarchical multimodal transformers for multi-page docvqa. *arXiv preprint arXiv:2212.05935*, 2022.

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.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, 37:95095–95169, 2024a.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. *arXiv preprint arXiv:2409.07429*, 2024b.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *Advances in Neural Information Processing Systems*, 38:17577–17604, 2026.

Zhiling Yan, Dingjie Song, Hanrong Zhang, Wei Liang, Yuxuan Zhang, Yutong Dai, Lifang He, Philip S Yu, Ran Xu, Xiang Li, et al. Openskill: Open-world self-evolution for llm agents. *arXiv preprint arXiv:2606.06741*, 2026.

Cheng Yang, Chufan Shi, Yaxin Liu, Bo Shui, Junjie Wang, Mohan Jing, Linran Xu, Xinyu Zhu, Siheng Li, Yuxiang Zhang, et al. Chartmimic: Evaluating lmm's cross-modal reasoning capability via chart-to-code generation. In *International Conference on Learning Representations*, volume 2025, pp. 26590–26646, 2025.

Cheng Yang, Xuemeng Yang, Licheng Wen, Daocheng Fu, Jianbiao Mei, Rong Wu, Pinlong Cai, Yufan Shen, Nianchen Deng, Jia Xu, et al. Towards self-evolving agents: Enabling autonomy through interactive experience refinement. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 30424–30451, 2026.

Yida Yin, Harish Krishnakumar, Chung Peng Lee, Boya Zeng, Wenhao Chai, Shengbang Tong, Wenhu Chen, Hu Xu, Xingyu Fu, Gabriel Sarch, et al. Worldbench: A challenging and visually diverse multimodal reasoning benchmark. *arXiv preprint arXiv:2606.06538*, 2026.

Siyu Yuan, Kaitao Song, Jiangjie Chen, Xu Tan, Dongsheng Li, and Deqing Yang. Evoagent: Towards automatic multi-agent generation via evolutionary algorithms. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6192–6217, 2025.

Ziyun Zeng, Hang Hua, and Jiebo Luo. Mira: Multimodal iterative reasoning agent for image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9563–9573, 2026a.

Ziyun Zeng, Hang Hua, Bocheng Zou, Mu Cai, Rogerio Feris, and Jiebo Luo. Mementogui: Learning agentic multimodal memory control for long-horizon gui agents. *arXiv preprint arXiv:2605.18652*, 2026b.

Boyang Zhang, Sebastián G Acosta, Preston Carlson, Sacha Bron, Pierre-Loïc Doulcet, Daniel B Ospina, and Simon Suo. Parsebench: A document parsing benchmark for ai agents. *arXiv preprint arXiv:2604.08538*, 2026a.

Guibin Zhang, Muxin Fu, Kun Wang, Frank Wan, Miao Yu, and Shuicheng Yan. G-memory: Tracing hierarchical memory for multi-agent systems. *Advances in Neural Information Processing Systems*, 38:12988–13018, 2026b.

Wenqi Zhang, Ke Tang, Hai Wu, Mengna Wang, Yongliang Shen, Guiyang Hou, Zeqi Tan, Peng Li, Yueting Zhuang, and Weiming Lu. Agent-pro: Learning to evolve via policy-level reflection and optimization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5348–5375, 2024.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

Boyuan Zheng, Michael Y Fatemi, Xiaolong Jin, Zora Zhiruo Wang, Apurva Gandhi, Yueqi Song, Yu Gu, Jayanth Srinivasa, Gaowen Liu, Graham Neubig, et al. Skillweaver: Web agents can self-improve by discovering and honing skills. *arXiv preprint arXiv:2504.07079*, 2025.

Longtao Zheng, Rundong Wang, Xinrun Wang, and Bo An. Synapse: Trajectory-as-exemplar prompting with memory for computer control. In *International Conference on Learning Representations*, volume 2024, pp. 19036–19066, 2024.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 19724–19731, 2024.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning acting and planning in language models. *arXiv preprint arXiv:2310.04406*, 2023.

Jiawei Zhou, Chi Zhang, Xiang Feng, Qiming Zhang, Haibo Qiu, Lihuo He, Dengpan Ye, Xinbo Gao, and Jing Zhang. Omni-i2c: A holistic benchmark for high-fidelity image-to-code generation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 37761–37798, 2026.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. Language agents as optimizable graphs. *arXiv preprint arXiv:2402.16823*, 2024.

A Data Construction Details

A.1 Source Tasks and Data Collection

Training tasks are drawn from MathNet (Alshammari et al., 2026), MathV360K (Shi et al., 2024), ChartQA (Masry et al., 2022), InfoVQA (Mathew et al., 2022), DocVQA (Mathew et al., 2021), and ChartNet (Kondic et al., 2026). They cover three domains: visual mathematics, document understanding, and reasoning over charts and infographics. The source tasks provide the questions and images on which we collect multi-agent system (MAS) executions. Memory updates, lessons, and tree-operation targets are subsequently generated from these execution records.

We collect trajectories using CAMEL (Li et al., 2023), Codex (OpenAI, 2026a), and DeepSeek-Harness (DeepSeek-AI, 2026). Qwen3.8-27B (Qwen Team, 2026b) generates initial memory updates and operation demonstrations. Qwen3.8-Flash-Next (Qwen Team, 2026a) selectively revises unsuccessful updates, improves lesson quality, and corrects tree-operation targets. The accepted targets provide separate supervision for the Memory Controller and Tree Self-Organizer.

A.2 Replay-Verified Joint Memory Transformations

Joint text–visual targets. Given the question, images, previous memory, latest attempt, and correctness feedback, Qwen3.8-27B proposes a complete memory state

$$m' = (\ell', g', p', b'), \tag{5}$$

where ℓ' is guidance text, g' specifies visual use, and p' and b' identify a source image and normalized bounding box. When $g' = 0$, the visual-reference fields are null. The target jointly specifies textual guidance and its associated visual evidence. The evaluator uses reference answers to assess attempts and produce correctness feedback. Qwen3.8-Flash-Next revises updates that fail to support successful solving and refines lessons where needed.

Repair and compression. Replay compares the previous and proposed memory states with the same MAS host. Repair transformations are retained when the new memory turns a failed attempt into a successful one. Compress transformations preserve successful solving while shortening text without increasing crop area:

$$\text{tokens}(\ell') < \text{tokens}(\ell), \qquad \text{area}(b') \leq \text{area}(b). \tag{6}$$

Here ℓ and b denote the previous text and visual region; crop area is normalized relative to the source image and is zero when no visual region is retained. The original task images remain available during replay. The final accepted memory, including any teacher revision, is the target selected through replay.

Each training example pairs the update inputs with the accepted memory JSON. The dataset contains approximately 25K memory transformations. Multiple transformations may originate from one source question, so this count denotes supervision examples rather than independent tasks. Replay validates the observed memory transition on the source task, not a guarantee of improvement on every subsequent task.

A.3 Tree Organization and Retrieval Demonstrations

We derive seed lessons from the collected controller memories and organize them into small memory banks. Historical lessons and retrieval queries are separated within each bank. Qwen3.8-27B

produces demonstrations of placement, merging, splitting, consolidation, and retrieval; Qwen3.8-Flash-Next selectively revises the operation targets.

The final targets are checked for schema validity, node references, partition coverage, selection constraints, and provenance. Valid empty retrievals are retained when no candidate applies. These checks establish structural validity; tree-operation demonstrations are not individually verified through downstream MAS replay. The dataset contains approximately 6.5K operation demonstrations, which count supervised operations rather than seed lessons or independent questions.

Training and bank construction. The supervision datasets train the two memory models. The benchmark-derived construction questions described in Appendix C are subsequently used to build external experience banks with the models fixed. These construction questions add memory content without updating model parameters. Evaluation uses separate questions and frozen historical banks.

B Training the Two Memory Policies

We independently fine-tune two Qwen3.5-2B models on the datasets in Section 3. For a policy P_ψ and its operation dataset $\mathcal{D}_P$, supervised fine-tuning minimizes the target-sequence loss

$$\mathcal{L}_P(\psi) = \mathbb{E}_{(z,y)\sim\mathcal{D}_P}\left[-\sum_{j=1}^{|y|}\log p_\psi(y_j \mid z, y_{<j})\right], \tag{7}$$

where z contains the operation prompt and its available inputs, and y is the validated memory or operation JSON. Controller targets cover repair and compression of joint text–visual memory. Tree targets cover batch placement, merging, splitting, consolidation, and retrieval; only retrieval examples include images.

Both models use LoRA with rank 16 and scaling factor 32, targeting linear layers while freezing the visual encoder and aligner. Training uses three epochs, a learning rate of 10^{-4}, maximum sequence length 8,192, per-device batch size 2, and gradient accumulation over 8 steps. Checkpoints are selected by validation loss: step 1,000 for the controller and step 258 for the tree memory self-organizer.

Training and deployment interfaces. Offline training-data construction uses correctness feedback to generate and screen memory targets. During evaluation with Codex and DeepSeek-Harness, MC revises memory from the question, images, existing memory, answer, and execution trace. The joint update supervises text, visual use, and the source image region together. Adaptive visual injection controls which stored visual memory enters the next solver input; omitting visual memory from an input does not delete it from the stored state.

C Experimental Protocol and Data Splits

Construction and evaluation sets. Table 8 lists the benchmark-specific question counts. Experiments that use historical memory share an initial set of 1,000 construction questions: 10 from DocVQA2026, 90 from MP-DocVQA, and 100 from each of the other nine benchmarks. Each memory setting builds its bank from the corresponding MAS execution records, or reuses the specified donor bank in transfer experiments. These questions construct memory content without updating model parameters. No Memory and the question-level ablations do not use a historical bank. The standard evaluation split contains 4,538 questions across eleven benchmarks; experiments reporting a subset of benchmarks use the corresponding evaluation subsets. Construction and evaluation are project-defined splits and are disjoint. For ReasonMap, questions are grouped by their source map before splitting: a map is assigned either to construction or to evaluation, never to both. This map-level separation also applies when adding the second construction set. Construction counts refer to source questions, not the number of accepted memory episodes.

Cross-harness memory evolution. The first construction stage uses the same 1,000 questions as the other historical-memory experiments. A second, non-overlapping set of 1,000 questions is used by the other harness to use and evolve the original bank. Table 8 gives the distribution for

Table 8: Construction and evaluation questions by benchmark. Initial construction is shared by historical-memory experiments and the first stage of cross-harness memory evolution. Additional construction is used only for memory evolution; its evaluation set excludes both construction sets.

Benchmark	Construction		Evaluation	
	Initial	Additional	Standard	Evolution
DocVQA2026	10	0	70	70
MP-DocVQA	90	100	500	500
ParseBench	100	50	468	418
Vision2Code	100	200	500	500
Omni-I2C	100	150	500	500
ChartMimic	100	0	500	500
MATH-Vision	100	100	500	500
WeMath	100	100	500	500
WorldBench	100	210	500	500
ReasonMap	100	40	212	162
BabyVision	100	50	288	238
Total	**1,000**	**1,000**	**4,538**	**4,388**

both stages. The evaluation split contains 4,388 questions and excludes both construction sets. Codex and DeepSeek-Harness exchange the roles of initial construction and subsequent evolution, with Qwen3.8-27B and the two trained 2B memory models fixed. For each evaluated harness, the original and evolved banks are compared on identical questions with one historical retrieval and a single solving attempt, without question-level memory updates. Table 3 follows the same single-attempt protocol as Table 1; its evaluation split additionally excludes the second construction set.

Evaluation controls. Model parameters and historical banks remain frozen during evaluation. The main comparison and all evaluations with historical memory retrieval use a single solving attempt per question, with question-level memory updates disabled. Each attempt includes the harness's standard internal agent and tool interactions. Where enabled in other experiments, question-level updates affect only the current question and are not written back to the bank. Reference answers and official scores are used only for offline evaluation and do not enter memory updates, stopping decisions, or answer selection. We report the last completed answer. Cross-backbone and cross-harness transfer compare one solve with and without the donor bank, with question-level updates disabled. No Memory makes a single solving attempt in every experiment, including the question-level controller comparison, multimodal ablation, and stronger-solver comparison. The question-level controller comparison and multimodal ablation disable historical retrieval and allow up to five solving attempts for memory-enabled settings, including the initial attempt. The system may stop earlier when it judges the current answer ready to return. Their comparisons against No Memory measure memory-guided refinement as a whole, including additional attempts. Memory-enabled settings in the multimodal ablation share the attempt budget. Original task images remain available in the multimodal ablation. Results may be reused across experiments only when the samples, harness, backbone, memory configuration, bank, and inference protocol match; a different evaluation subset requires recomputing scores from the corresponding predictions.

Score and cost reporting. We use each benchmark's evaluation metric and give equal weight to each benchmark when computing macro-average scores. Time and token usage are reported separately for MAS execution and memory operations. For each comparison, let T_0 and N_0 be the corresponding No Memory MAS time and token usage under the same harness, backbone, evaluation set, and baseline protocol. We normalize each cost component c as

$$\widetilde{T}_c = \frac{T_c}{T_0}, \qquad \widetilde{N}_c = \frac{N_c}{N_0}, \qquad c \in \{\mathrm{MAS}, \mathrm{Memory}\}. \tag{8}$$

Thus, No Memory MAS time and token usage are each $1\times$, and its memory-operation costs are zero. Memory costs use the same MAS denominators rather than a separate memory baseline.

Hardware. All experiments are run using NVIDIA H100 GPUs. GPT-5.6-Luna is accessed through its API.

D Additional Performance and Cost Comparisons

Figures 4–5 extend Figure 3 using the same main-results measurements. With Gemma4-31B, backbone-sized EpiCon improves the eleven-benchmark mean over No Memory by 7.0 points on Codex and 2.7 points on DeepSeek-Harness; the trained 2B variant gains 4.2 and 1.7 points with lower total time than the backbone-sized variant. Figure 5 compares token usage on DeepSeek-Harness with the two backbones. Relative to backbone-sized EpiCon, the trained 2B variant uses 25.1% fewer total tokens with Qwen3.8-27B and 16.8% fewer with Gemma4-31B, while yielding lower mean benchmark scores. These results show the performance–token trade-off of the two trained 2B memory models across backbones.

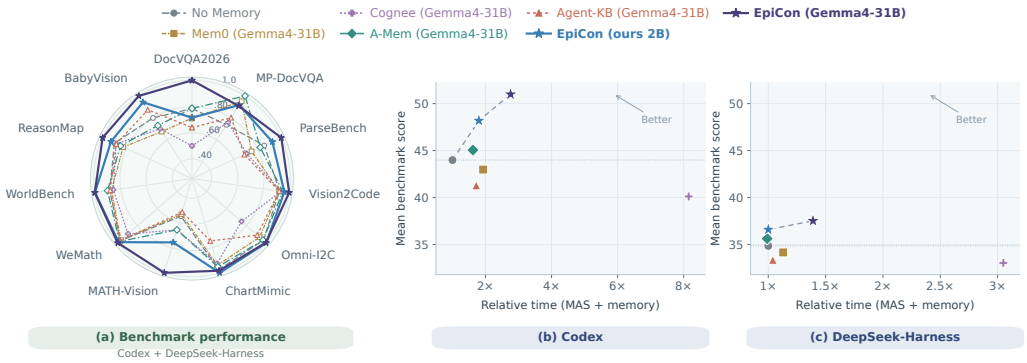

Figure 4: Benchmark profiles and performance–time trade-offs with Gemma4-31B, using the results in Table 1. (a) Scores are averaged across Codex and DeepSeek-Harness, then divided by the best method mean for each benchmark; the radial axis starts at 0.25, as in Figure 3. (b) and (c) show performance as the equally weighted mean across eleven benchmarks. Total time sums the normalized MAS and memory costs within each harness. Dashed lines connect No Memory, EpiCon (ours 2B), and EpiCon (Gemma4-31B) as visual guides, not Pareto fronts.

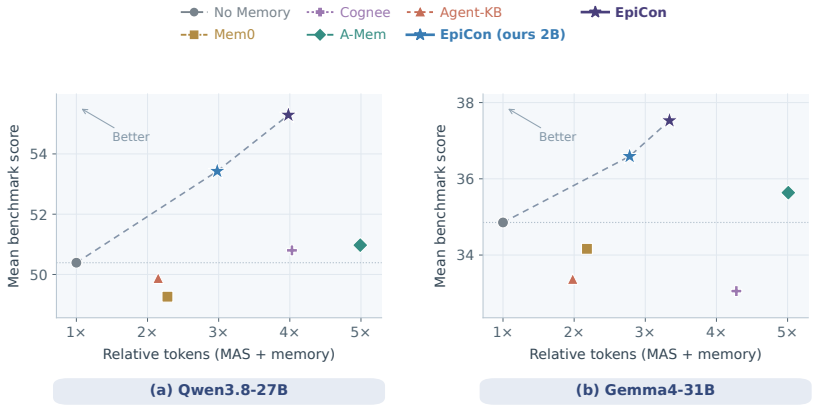

Figure 5: Performance–token trade-offs on DeepSeek-Harness with (a) Qwen3.8-27B and (b) Gemma4-31B. Performance is the equally weighted mean across eleven benchmarks in Table 1. Total token usage sums MAS and memory tokens, normalized by the corresponding No Memory MAS usage for each backbone. Memory methods use the backbone named below each panel, except EpiCon (ours 2B), which uses two independently trained 2B memory models. The panels share the token scale and use separate score ranges. Dashed lines connect No Memory and the two EpiCon variants as visual guides, not Pareto fronts.