

MUSE: Evolving Grounded Multimodal Skills for Cross-Episode Agent Improvement

Anonymous ACL submission

Abstract

Multimodal agents can solve complex visual problems, but individual successes do not automatically become reusable procedures for later tasks. Recurring diagrams, layouts, and visual conventions provide an opportunity to accumulate such experience across episodes. We introduce MUSE (MULTimodal Skill Evolution), a framework that distills successful multimodal trajectories into reusable, image-conditioned perceptual-reasoning procedures. MUSE summarizes successful trajectories into structured textual profiles containing visual observation cues, reasoning procedures, and image-based answer verification checks. Each library is constructed from 20 labeled seed examples over 20 offline rounds and frozen before evaluation. Across four benchmarks and five frontier LVLs, the complete MUSE protocol outperforms the evaluated baselines, increasing GPT-5.4 average accuracy from 72.5% to 87.3% and reaching 91.8% on Claude-Sonnet-4.6. Removing visual guidance from the profiles lowers GPT-5.4/MathVista accuracy from 83.1% to 71.3%, despite retaining the input image. This ablation supports image-conditioned procedural guidance as an important component of reliable skill reuse.

1 Introduction

While multimodal large language models (MLLMs) are rapidly advancing, their failures in complex visual reasoning rarely stem from deficient symbolic logic, but rather from systematic visual misperceptions (Lu et al., 2023; Chen et al., 2024b; Cheng et al., 2025b). Faced with intricate visual scenes, models repeatedly stumble over recurring perceptual traps: missing subtle axis truncations, misinterpreting layout conventions, and grounding reasoning on erroneous visual artifacts. This indicates that the true bottleneck of multimodal reasoning lies in *how to perceive and verify*, whose successful trajectories encapsulate

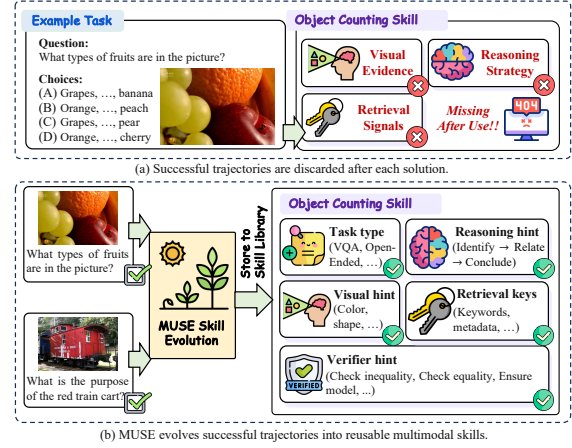


Figure 1: Motivation for multimodal skill evolution. Successful trajectories are usually discarded after solving an example; MUSE instead extracts visual evidence and reasoning strategies into reusable skill profiles.

vital cross-task invariants of visual attention allocation and spatial cue binding.

However, existing paradigms leave multimodal systems unable to retain useful experience across tasks, and worse, naively borrowing pure-text skill reuse strategies directly precipitates catastrophic failure in the visual domain. While pure-text skills are symbolically self-contained, visual reasoning is deeply embodied and structurally sensitive; devoid of pixel-grounded visual invariants, abstract textual procedures lead to errors in image-grounded reasoning. Our diagnostic experiments reveal these limitations of text-only skill reuse: naively transferring text-abstracted skills to new images induces aggressive visual hallucinations, collapsing accuracy from 83.1% to 71.3%, substantially lower than the 79.0% zero-reuse baseline. This dramatic divergence proves that multimodal agents cannot inherit text-only skill frameworks, establishing that reusable visual experience strictly demands an intrinsically grounded multimodal formulation (Figure 1).

To resolve this, we introduce MUSE (MULTImodal Skill Evolution), a paradigm that transforms episodic visual successes into reusable, image-conditioned *perceptual-reasoning procedures*. An evolved MUSE skill is a structured textual profile with visual, reasoning, and verification guidance, plus applicability and retrieval metadata: it guides *observation of visual cues and spatial conventions, reasoning over image structure, and image-based answer verification*. To prevent over-generalization during skill reuse, MUSE institutes a decoupled dual-agent architecture: the *multimodal reasoning agent* focuses on frozen skill-guided perception-deduction execution, while the *multimodal skill evolution agent* orchestrates seed qualification, spatial feature abstraction, and trust-gated reuse. This architecture guarantees that an evolved skill is admitted only when fully corroborated by visual evidence and ratified by image-based answer verification.

We evaluate MUSE on four multimodal reasoning benchmarks (MATHVISTA, M3COT, MATHVERSE, and MMMU-PRO) using five advanced LVLMS. MUSE improves accuracy over the evaluated baselines across all five models. For GPT-5.4, average accuracy increases from 72.5% to 87.3%, including a 31.3-percentage-point improvement on MATHVERSE. Claude-Sonnet-4.6 achieves an average accuracy of 91.8% with MUSE. We further evaluate two open-source 8B models (InternVL3.5 and Qwen3-VL) and observe improvements of up to 10.8 percentage points without parameter updates. These results suggest that structured skill reuse can improve visual reasoning across different model backbones.

Our principal contributions are threefold:

- We study the limitations of text-only skill reuse in multimodal reasoning and introduce MUSE, a framework for extracting reusable, image-conditioned reasoning procedures from successful multimodal trajectories.
- We design a structured skill profile with visual, reasoning, and verification guidance, plus applicability and retrieval metadata, paired with a trust-dependent reuse gate that insulates the agent from destructive over-generalization.
- Across 7 LVLMS, MUSE achieves massive improvements, while providing causal empirical proof that explicit visual grounding is the non-negotiable determinant for lifelong multimodal

agent evolution.

2 Related Work

Recent advances enhance multimodal reasoning through test-time computation, self-reflection, and vision-grounded verification (Chen et al., 2024a; Madaan et al., 2023; Cheng et al., 2025a; Wang et al., 2025a; Sun et al., 2025). These approaches motivate improvements within a reasoning rollout. Our focus is complementary: how to represent successful observation and verification strategies for reuse across independent tasks. We distinguish improving a current solution from constructing an explicit library of reusable procedures.

A natural remedy is equipping agents with persistent memory, ranging from retrieval-augmented generation (Lewis et al., 2020) and generative interaction logs (Park et al., 2023) to structured trajectory archives (Xu et al., 2025; Xiong et al., 2025). Memory mechanisms provide a substrate for retaining experience. We study a particular representation within that design space: a textual profile that separately describes applicability, visual observation, reasoning, and verification. The AMem comparison evaluates our combined admission-and-abstraction protocol against raw-trajectory retrieval. This comparison is limited to the raw-trajectory retrieval implementation used in our experiments. Our ablations show that removing visual guidance from evolved skills can reduce accuracy below Direct, motivating image-conditioned procedural guidance.

MUSE abstracts successful multimodal trajectories into image-conditioned *perceptual-reasoning procedures*, unifying visual observation directives, reasoning procedures, and image-conditioned verification. Its skills guide a backbone through language; their multimodal character comes from the source experience and their dependence on observations from the current image.

3 Multimodal Skill Evolution (MUSE)

MUSE explores how multimodal agents can transform episodic visual successes into reusable, image-conditioned *perceptual-reasoning procedures*. Distilled from multimodal trajectories, these image-conditioned skills integrate visual guidance, structured reasoning, and verification into a coherent loop: *where to look, what evidence to extract and how to reason from it, and how to verify conclusions against the current image*.

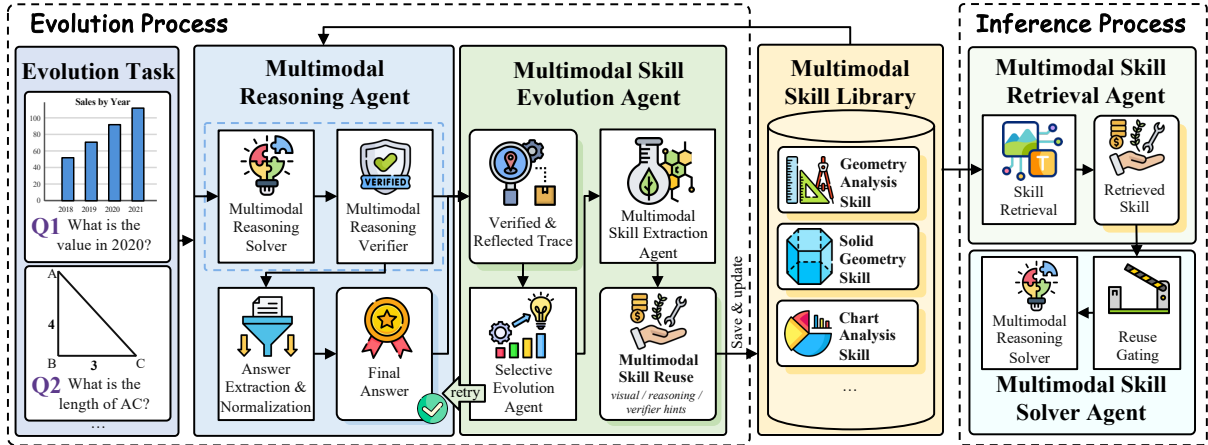


Figure 2: Overview of MUSE as multimodal skill evolution.

3.1 Problem Formulation

We consider multimodal reasoning instances $x_i = (I_i, q_i, c_i, m_i)$, where I_i is visual content, q_i is a question, c_i contains optional choices, and m_i contains metadata such as source, scene, question type, answer type, and skill tags when available. A system returns a prediction $\hat{y}_i$. The seed set $\mathcal{D}_{\text{seed}}$ and evaluation set $\mathcal{D}_{\text{eval}}$ are disjoint. A separate M3CoT validation subset supports threshold selection. The library and reuse-support state are frozen before evaluation.

3.2 MUSE Evolution Process

As shown in Figure 2, MUSE evolves skills in three stages. First, a multimodal reasoning agent solves seed instances and records structured trajectories of visual observations, reasoning steps, and verification outcomes. Second, selective admission retains only error-free, independently solved (not skill-guided), answer-correct trajectories. Third, a skill evolution agent distills each admitted trajectory into reusable skills specifying *what to look at*, *how to reason*, and *how to verify*, without copying the seed answer. Our main setting initializes 20 skills in the first round, refines them offline, and freezes the resulting set $\mathcal{S}$ before evaluation.

Prompt-level iteration. Each evolution round applies the observation, reasoning, verifier, bounded-retry, and extraction templates in Appendix C to seed experience. The first round creates skill profiles; later rounds refine them through the same extraction interface, conditioned on the current profile and preserving its schema. Refinement updates applicability, visual, reasoning, verification, and retrieval fields without adding another solver. Only

independent trajectories satisfying Equation 1 provide new evidence for profile updates; skill-guided solves are assessed for reuse support through a separate gate. Appendix A.4 details the round-level data flow and both gates.

3.2.1 Multimodal Reasoning Agent

MUSE uses the same solver for evolution and inference. Offline, the verifier and skill-evolution modules validate and summarize seed experience. At inference, the multimodal skill retrieval agent retrieves candidate profiles from the frozen library, and the multimodal skill solver agent performs skill-guided solving and reuse gating, as shown in Figure 2.

Multimodal Reasoning Solver receives the image I_i and question q_i , then performs structured visual observation: it selects an observation strategy (e.g., OCR extraction, object detection, diagram parsing, chart-axis reading, or spatial-relation analysis) based on the visual content type, and produces a set of evidence statements grounded in specific image regions. The solver then reasons over the collected evidence to derive a candidate answer¹.

Multimodal Reasoning Verifier. During offline evolution, the verifier re-examines the image against the solver’s prediction. Its image-conditioned interface produces both a decision and a confidence score for online reuse gating by checking visual consistency, answer format (e.g., numeric, multiple-choice, or free-form text), and other potential inconsistencies. When the prediction is rejected, the solver performs a second reasoning pass to revise its answer.

¹If using a skill, see the inference process for details.

The resulting trajectory is $\tau_i = (x_i, o_i, r_i, v_i, \hat{y}_i)$, where o_i , r_i , v_i , and $\hat{y}_i$ denote the observations, reasoning summary, verifier feedback, and final prediction, respectively. This trajectory provides the basis for skill evolution: o_i captures the visual evidence examined, r_i records the reasoning operation applied to it, and v_i preserves image-grounded constraints that help prevent a skill from being reused for an incompatible visual pattern.

3.2.2 Multimodal Skill Evolution Agent

Skills in MUSE are structured textual profiles derived from multimodal experience, with image-conditioned applicability. Their visual fields guide the backbone or available perception tools to inspect and extract evidence from the current image, while their reasoning and verifier fields guide analysis and image-based checking. The profile specifies these operations without itself implementing a separate detector or perception module. They are designed to capture transferable observation and reasoning procedures while omitting instance-specific answers.

Selective Evolution Agent Not every correct trajectory should become a skill. Admission excludes execution failures and skill-guided traces, and requires seed-answer matching. These checks reduce unsuitable experience but do not prove that every admitted reasoning trace is correct. A trajectory τ_i is evolved only if:

$$\text{admit}(\tau_i) = \mathbb{I}[\text{save} = 1 \wedge e_i = 0 \wedge u_i = 0 \wedge \text{match}(\hat{y}_i, a_i^*)], \quad (1)$$

where $e_i=0$ indicates error-free execution, $u_i=0$ indicates an answer produced without an existing skill, and $\text{match}(\hat{y}_i, a_i^*)$ denotes agreement with the gold answer under task-normalized comparison. Seed gold answers are used *only* in the offline Boolean gate $\mathbb{I}$, never as gold-label inputs to the skill evolver or as skill-field content. This supervision is confined to seed construction; evaluation answers remain inaccessible at test time. For rejected seeds with error-free execution, MUSE permits up to two reflection retries, each guided by the verifier’s rejection reason. The retried trajectory is retained only if its final attempt passes the gate.

Multimodal Skill Extraction Agent For each selected trajectory τ_i , the skill evolver performs a structured extraction that preserves the multimodal grounding of the original solving process. It receives the task description (without the gold answer), the solver’s visual observation experience

o_i , reasoning summary r_i , verifier feedback v_i , and the final prediction $\hat{y}_i$. From these, it summarizes a reusable skill profile by answering five questions: (1) **When should this skill apply?** The evolver identifies the visual and task conditions (e.g., “bar chart with categorical x-axis and a comparison question”) and writes them as applicability conditions ϕ_i . (2) **What should the solver look at?** It extracts the multimodal observation strategy from o_i , which image regions to attend to, what information to extract, and generalizes it into visual hints h_i^v . (3) **How should the solver reason?** It distills the reasoning pattern from r_i into a transferable reasoning procedure h_i^r , stripping away instance-specific details. (4) **How should the verifier check?** It converts the verification logic from v_i into reusable verifier constraints h_i^z (e.g., “answer must be one of the axis labels”). (5) **How should this skill be found later?** It generates retrieval descriptors k_i (task type, visual scene, key concepts) for the retrieval router.

The resulting skill profile is:

$$s_i = \text{Evolve}(\tau_i) = (\phi_i, h_i^v, h_i^r, h_i^z, k_i). \quad (2)$$

Each profile describes a reusable procedure and excludes the seed answer. Its fields describe how to look at, analyze, and verify a new image; they are not themselves executable visual programs or stored crops, masks, or fixed pixel correspondences. At reuse time, ϕ_i describes relevant visual configurations, h_i^v directs the solver toward evidence to inspect and extract, h_i^r guides analysis of those observations, and h_i^z supplies checks against the current image. The backbone or tools carry out the visual operations, while the profile organizes when and how they should be used.

Skill Reuse Qualification is the label-free Reuse Gating step in Figure 2, based on image-grounded checking outputs. An accepted reuse is a skill-use decision that passes the following gate: the checking decision is accept, the answer can be canonicalized, and the reported confidence meets a trust-dependent threshold. Gate acceptance does not establish gold-label correctness:

$$\begin{aligned} \text{Accept}(s_i, x_j) = \mathbb{I}[v_{ij} = \text{accept} \\ \wedge b_{ij}^{\text{can}} = 1 \\ \wedge c_{ij}^{\text{ver}} \geq \theta(\text{trust}_i)], \end{aligned} \quad (3)$$

where v_{ij} denotes the reuse-check decision, b_{ij}^{can} indicates successful answer canonicalization, and

c_{ij}^{ver} denotes verification confidence. Trust aggregates accepted reuse decisions from offline evolution and is frozen before evaluation, enabling inference-time reuse gating without access to evaluation labels.

3.3 MUSE Inference Process

At inference time, MUSE follows the inference process in Figure 2: skill retrieval, skill-guided solving, reuse gating, and answer extraction and normalization. The skill library and trust state remain frozen throughout evaluation; evaluation instances do not create or update skills.

Multimodal Skill Retrieval Agent. Given the current image, question, optional choices, and task metadata, the retrieval agent scores stored skills using metadata, lexical, scene, shared-skill, history, and generalization-risk matches. It retrieves candidate profiles for the skill solver agent:

$$C(x_j) = \text{Top-K}\{s_i \mid R_{\text{cand}}(s_i, x_j) \geq \tau\}. \quad (4)$$

The threshold $\tau = 4.5$ is selected on the M3CoT validation set and fixed before evaluation. Retrieval uses the frozen skill library and does not require evaluation labels.

Multimodal Skill Solver Agent. The skill solver agent receives the task and retrieved profiles and checks their applicability conditions ϕ_i against the current image and question. For an applicable profile, its visual hints h_i^v guide evidence inspection, reasoning hints h_i^r guide deduction, and verifier hints h_i^z guide image-grounded checking. Reuse Gating applies Eq. 3 to qualify the skill-guided result. Retrieval alone does not establish successful reuse or answer correctness. If no candidate is applicable or no skill-guided result passes the gate, the solver proceeds without accepted skill guidance. The resulting prediction is passed to answer extraction and normalization. Appendix A.4 describes the boundary between offline construction and this inference process.

4 Experiments

4.1 Experimental Setup

We evaluate MUSE with a 20-skill library built over 20 offline evolution rounds from 20 labeled randomly sampled seed examples and frozen before inference. Each of the three runs independently rebuilds the skill library. Seed and evaluation examples are completely disjoint, and evaluation instances cannot create or update skills.

We use four benchmarks: MATHVISTA, M3CoT, MATHVERSE, and MMMU-PRO. Full details are in Appendix A.

4.2 Baselines

Direct supplies only the task input to the tool-enabled backbone, which manages reasoning and tool use. **Self-Consistency** (Wang et al., 2023) uses $k = 5$ independent direct rollouts and selects the final answer by majority voting. **Reflexion** (Shinn et al., 2023; Madaan et al., 2023) adds one image-grounded critique-and-revise step without skill guidance. **AMem** (Xu et al., 2025) retrieves evolution-stage memory snippets. Details are in Appendix C. The main table reports 5 advanced LVLMS: GPT-5.4 (Team, 2026c), GPT-5.4-mini (Team, 2026c), Gemini-3-Flash (Team, 2026b), Claude-Sonnet-4.6 (Team, 2026a), and GLM-4.6-V (Team, 2026d). We additionally run a transfer check on two 8B open-source LVLMS, InternVL3.5-8B (Wang et al., 2025b) and Qwen3-VL-8B-Thinking (Bai et al., 2025).

4.3 Main Results

Multimodal skill evolution improves accuracy across the evaluated backbones. We evaluate MUSE across 5 frontier LVLMS and 4 challenging benchmarks. As demonstrated in Table 1, MUSE achieves the highest average accuracy among Direct solving, self-consistency, reflection, and text-memory baselines on every evaluated backbone. Notably, MUSE improves GPT-5.4’s average accuracy from 72.52% to 87.26%, with a 31.30-point improvement on MATHVERSE, and achieves 91.84% average accuracy with Claude-Sonnet-4.6. These results support the complete skill-evolution protocol under the evaluated settings; Appendix B provides the split robustness and retrieval controls supporting transfer of the learned procedures.

MUSE transfers to open-source LVLMS. To confirm that skill evolution is not limited to closed-source models, we additionally evaluate MUSE on M3CoT with two open-source LVLMS. As shown in Figure 3, MUSE improves InternVL3.5-8B and Qwen3-VL-8B-Thinking over Direct by 10.75% and 8.00%, respectively, and gives the best M3CoT accuracy among the compared methods. Appendix D gives the corresponding table. This result suggests that the skill-evolution mechanism can transfer beyond closed-source models when LVLMS can follow the structured skill profile.

Backbone	Method	MathVista	M3CoT	MathVerse	MMMU-Pro	Avg.
GPT-5.4	Direct	78.98	85.08	62.10	63.90	72.52
	Self-Consistency (Wang et al., 2023)	79.18	86.00	64.10	63.31	73.15
	Reflexion (Shinn et al., 2023)	75.20	82.75	76.80	59.92	73.67
	AMem (Xu et al., 2025)	79.29	84.42	58.30	63.81	71.46
	MUSE	83.11	91.42	93.40	81.10	87.26
GPT-5.4-mini	Direct	71.94	77.83	58.20	52.29	65.07
	Self-Consistency (Wang et al., 2023)	73.27	78.42	62.30	53.98	66.99
	Reflexion (Shinn et al., 2023)	65.92	74.58	73.40	49.66	65.89
	AMem (Xu et al., 2025)	71.71	78.08	65.97	52.54	67.08
	MUSE	73.98	86.83	86.50	72.20	79.88
Gemini-3-Flash	Direct	77.86	80.25	78.00	52.30	72.10
	Self-Consistency (Wang et al., 2023)	80.41	84.75	79.80	57.54	75.63
	Reflexion (Shinn et al., 2023)	54.69	66.83	79.00	32.12	58.16
	AMem (Xu et al., 2025)	78.67	81.25	78.30	52.20	72.61
	MUSE	84.39	86.75	82.50	67.23	80.22
Claude-Sonnet-4.6	Direct	81.49	86.67	81.90	67.65	79.43
	Self-Consistency (Wang et al., 2023)	81.33	87.83	82.90	68.81	80.22
	Reflexion (Shinn et al., 2023)	80.46	87.67	88.50	64.61	80.31
	AMem (Xu et al., 2025)	82.52	87.33	82.88	68.69	80.36
	MUSE	93.49	89.83	91.50	92.54	91.84
GLM-4.6-V	Direct	79.67	70.33	74.60	49.83	68.61
	Self-Consistency (Wang et al., 2023)	81.02	72.25	77.50	54.66	71.36
	Reflexion (Shinn et al., 2023)	78.96	72.75	77.80	47.79	69.33
	AMem (Xu et al., 2025)	78.65	71.67	74.40	50.09	68.70
	MUSE	85.29	84.67	79.40	60.54	77.48

Table 1: Main results on advanced LVLMS. Values are mean accuracies over three runs after answer normalization.

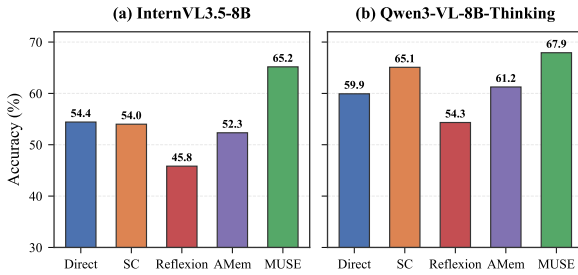


Figure 3: Results of M3CoT on open-source LVLMS, including InternVL3.5-8B and Qwen3-VL-8B-Thinking.

4.4 Diagnostic Analysis

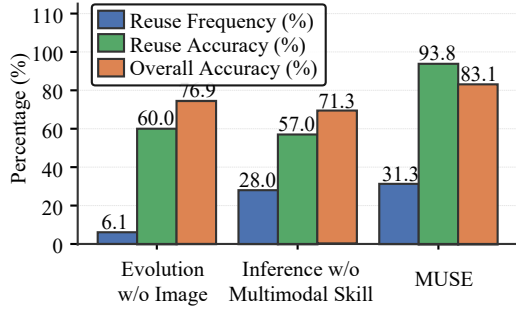
Visual guidance improves reliability in the grounding ablations. Ablations show that visual grounding is critical during both skill evolution and inference (Figure 4(a)). Removing images during evolution reduces the reuse frequency to 6.1%. During inference, **w/o multimodal skill** retrieves text-only skills without multimodal descriptions, although the solver still receives the image. The corresponding offline reuse frequency remains similar (28.0% vs. 31.3% for MUSE), but sharply reduces reuse accuracy from 93.8% to 57.0% and overall accuracy from 83.1% to 71.3%, falling below Direct (78.98%). Offline reuse frequency and evaluation accuracy describe different stages; frequent offline reuse alone is insufficient: without visual guidance, skill reuse can remain frequent

while becoming substantially less reliable. Manual inspection of 100 randomly sampled instances further confirms that removing visual guidance often induces visual hallucinations during reuse.

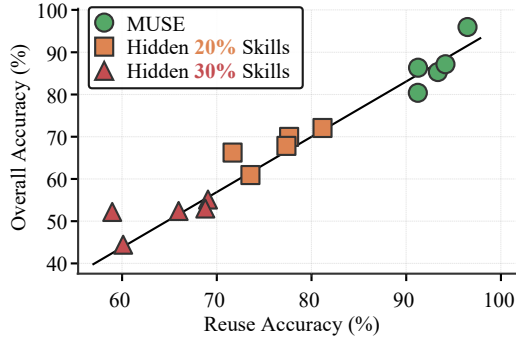
Reuse accuracy causally drives overall performance. Randomly hiding 20% or 30% of evolved skills creates a direct causal test: as reuse accuracy drops, the system’s ability to correctly apply the remaining skills declines. Overall accuracy falls proportionally along a tight linear relationship (Figure 4(b)). This controlled degradation reveals that performance gains stem not from skill quantity alone, but from the system’s learned capacity to identify and execute the right skill in each context. Evolution optimizes for usability: a skill contributes to overall performance only when the system knows when and how to invoke it.

4.5 Ablation Studies

Both multimodal skill content and evolution matter. We ablate two ingredients of MUSE: multimodal skill content and the evolution process. As shown in Figure 5(a), w/o MM retains text skills but removes multimodal skill information, whereas w/o Evol. uses the multimodal skills summarized in the first round and disables subsequent skill evolution. On MathVerse, w/o MM achieves 78.3%, w/o Evol. achieves 79.0%, and full



(a) Multimodal effect, removing multimodal information at either stage degrades reuse quality and overall performance.



(b) The correlation between the accuracy of the reuse skill and the overall accuracy.

Figure 4: Diagnostic analysis on MATHVISTA with GPT-5.4. Here, reuse frequency is the skill reuse rate during evolution; reuse accuracy is final-answer accuracy within the skill-use subset.

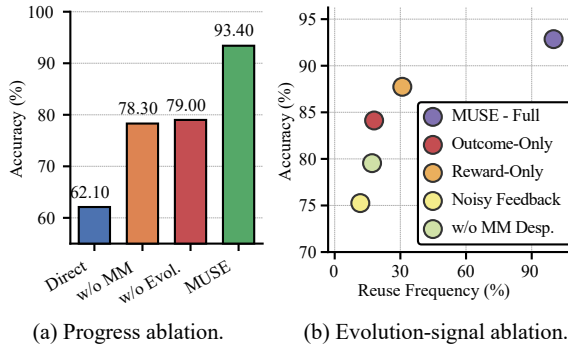
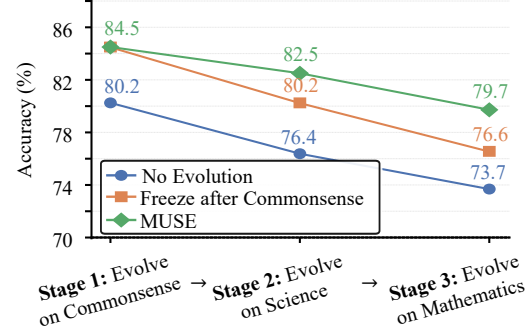


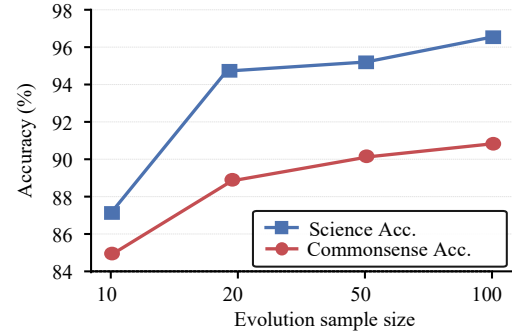
Figure 5: Ablation studies on MATHVERSE.

MUSE achieves 93.4%, compared with 62.1% for Direct. First-round multimodal skills improve over Direct, while subsequent evolution provides a further 14.4% gain. These results support the value of both multimodal skill content and evolution beyond the initial summarization round.

Skill evolution outperforms raw memory reuse. We compare MUSE with AMem, which retrieves raw trajectories without selective admission, evaluating admission and skill abstraction jointly (Appendix Table 3). AMem matches or underperforms Direct on three of four benchmarks, dropping 3.8%



(a) Cross-domain continual evolution of MUSE on different M3CoT domains (Commonsense → Science → Mathematics).



(b) Scaling evolution size of MUSE on M3CoT.

Figure 6: Continual skill evolution on M3CoT.

on MathVerse. By admitting verified solutions and abstracting them into skill profiles, MUSE surpasses AMem by 35.1% on MathVerse, 17.3 on MMMU-Pro, and 7.0 on M3CoT. Beyond this joint benefit, fully evolved skills outperform first-round skills of similar length (Figure 5(a)), demonstrating the value of iterative refinement beyond initial summarization.

Reliable skill reuse hinges on feedback quality and visual grounding. We compare MUSE with three evolution-feedback ablations: **Reward-only** (numerical correctness signals), **Outcome-only** (detailed feedback without numerical rewards), and **Noisy Feedback** (feedback randomly sampled from another instance). We further evaluate **w/o MM Desp.**, which omits multimodal descriptions during retrieval. All four ablations consistently degrade performance (Figure 5(b)), underscoring that effective multimodal skill reuse is precision-sensitive: it depends on reliable evolution signals and faithful visual grounding.

4.6 Continual Skill Evolution Analysis

Continual evolution improves performance across domain stages. We evaluate continual skill evolution on a M3CoT stream ordered Commonsense → Science → Mathematics, freezing the

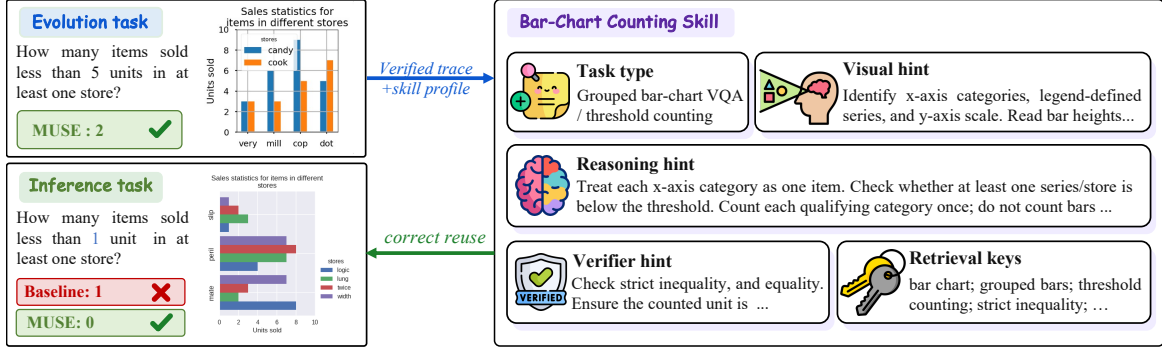


Figure 7: Example of an evolved multimodal skill profile in MUSE.

library at each stage boundary and disabling skill saving during evaluation. Figure 6(a) compares No Evolution, Freeze after Commonsense, and continual MUSE on each stage’s current domain. Continual MUSE achieves 84.5%, 82.5%, and 79.7%, respectively. On Science and Mathematics, it outperforms both No Evolution (76.4% and 73.7%) and Freeze after Commonsense (80.2% and 76.6%). These within-stage comparisons demonstrate the benefit of continued evolution; cross-stage differences also reflect domain difficulty.

Scaling evolution examples improves domain accuracy. We next vary the number of evolution examples from 10 to 100 to measure how the skill library scales under continual evolution. As shown in Figure 6(b), Science accuracy rises from 87.2% to 96.5%, while Commonsense accuracy rises from 84.9% to 90.9%. Most of the gain appears when moving from 10 to 20 examples, followed by steadier improvements as the library grows. This trend indicates that even a small evolved library can capture transferable visual-reasoning procedures, while additional examples mainly broaden skill coverage and reuse.

4.7 What Does an Evolved Skill Look Like?

An evolved multimodal skill is not a past solution trace, but a factorized profile (Figure 7). It decides when the skill applies, retrieval descriptors make it searchable, visual hints tell the solver what to inspect, reasoning hints specify the procedure, and verifier hints define what must be checked. This decomposition turns one successful trajectory into a reusable interface for later problems, providing a compact alternative to storing complete demonstrations; the retrieval controls in Appendix B further test whether the guidance matches the current task.

A skill is useful only if it can be matched and checked. Visual and verifier fields jointly act as a type-checking mechanism. A bar-chart skill should not fire on a pie chart: the retrieved procedure may sound plausible, but its visual evidence and verifier constraints no longer match the image. This explains full MUSE’s high final-answer accuracy (Figure 4(a)). Removing multimodal skill fields can cause confident use of unsuitable skills.

A skill stores procedures, not answers. The reasoning field describes reusable operations (“read the y-axis value at the labeled intersection”), excluding seed-specific answers (“42”). Profiles guide current-image analysis without stored answers. The agent must inspect the image and follow the checking guidance before answering.

The skill explains the grounding ablations. This factorization helps interpret the effect of visual guidance at both stages. Without images during evolution, visual and verifier fields cannot encode what to look for; without multimodal skill fields during reasoning, retrieved profiles cannot be validated against the current visual input. These support using image-conditioned guidance during both construction and reuse.

5 Conclusion

We presented MUSE, which distills successful multimodal trajectories into reusable, image-conditioned perceptual-reasoning procedures, addressing limited retention of successful reasoning strategies across tasks. Extensive evaluations across frontier LVLMs confirm that MUSE decisively outperforms reflection and memory baselines, while diagnostic ablations demonstrate that explicit visual grounding is essential to prevent severe hallucinations and enable genuine cross-episode competence accumulation.

Limitations

While MUSE demonstrates robust cross-episode transfer, it has several limitations: (1) initializing the skill library currently requires a small labeled seed set to guarantee admission fidelity; (2) skill-guided observation depends on the underlying backbone’s visual resolution and instruction-following capability; and (3) scaling to open-ended lifelong streams will require dynamic library consolidation and pruning to manage evolving skill growth over indefinite horizons.

References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025. Qwen3-v1 technical report. *arXiv preprint arXiv:2511.21631*.

Qiguang Chen, Libo Qin, Jiaqi Wang, Jinxuan Zhou, and Wanxiang Che. 2024a. Unlocking the capabilities of thought: A reasoning boundary framework to quantify and optimize chain-of-thought. *Advances in Neural Information Processing Systems*, 37:54872–54904.

Qiguang Chen, Libo Qin, Jin Zhang, Zhi Chen, Xiao Xu, and Wanxiang Che. 2024b. **M³CoT: A novel benchmark for multi-domain multi-step multi-modal chain-of-thought**. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Kanzhi Cheng, Li YanTao, Fangzhi Xu, Jianbing Zhang, Hao Zhou, and Yang Liu. 2025a. Vision-language models can self-improve reasoning via reflection. 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)*, pages 8876–8892.

Zihui Cheng, Qiguang Chen, Jin Zhang, Hao Fei, Xiaocheng Feng, Wanxiang Che, Min Li, and Libo Qin. 2025b. Comt: A novel benchmark for chain of multi-modal thought on large vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23678–23686.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023. **Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts**. *arXiv preprint arXiv:2310.02255*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean

Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. **Self-refine: Iterative refinement with self-feedback**. *arXiv preprint arXiv:2303.17651*.

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

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

Linzhuang Sun, Hao Liang, Jingxuan Wei, Bihui Yu, Tianpeng Li, Fan Yang, Zenan Zhou, and Wentao Zhang. 2025. Mm-verify: Enhancing multimodal reasoning with chain-of-thought verification. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14100–14115.

Anthropic Team. 2026a. **Claude Sonnet 4.6 model card**.

Google DeepMind Team. 2026b. **Gemini-3-Flash technical report**.

OpenAI Team. 2026c. **GPT-5.4 technical report**.

Zhipu AI Team. 2026d. **GLM-4.6-V technical report**.

Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhui Chen. 2025a. **VL-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning**. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, and 56 others. 2025b. **InternV3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency**. *arXiv preprint arXiv:2508.18265*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. **Self-consistency improves chain of thought reasoning in language models**. In *International Conference on Learning Representations*.

Zidi Xiong, Yuping Lin, Wenya Xie, Pengfei He, Zirui Liu, Jiliang Tang, Himabindu Lakkaraju, and Zhen Xiang. 2025. How memory management impacts llm agents: An empirical study of experience-following behavior. *arXiv preprint arXiv:2505.16067*.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025. **A-mem: Agentic memory for LLM agents**. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhui Chen, and Graham Neubig. 2024. **MMMU-Pro: A more robust multi-discipline multimodal understanding benchmark**. *arXiv preprint arXiv:2409.02813*.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Peng Gao, and Hongsheng Li. 2024. [Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems?](#) *arXiv preprint arXiv:2403.14624*.

Appendix

A Experimental Protocol

A.1 Experimental Controls

Data splits. All backbones and compared methods share the same fixed random seed and data splits within each benchmark. Each library uses 20 randomly sampled, labeled seed examples for evolution. Benchmark-specific settings are as follows:

- **MathVista (Lu et al., 2023):** 20 examples from the 1,000-example *testmini* split for evolution and the remaining 980 for evaluation.
- **M3CoT (Chen et al., 2024b):** 20 validation examples for evolution and 1,300 test examples for evaluation.
- **MathVerse (Zhang et al., 2024):** 20 seed examples and 200 disjoint evaluation examples per version. We run the five aligned versions (Text Dominant, Text Lite, Vision Intensive, Vision Dominant, and Vision Only) separately to preserve modality alignment.
- **MMMU-Pro (Yue et al., 2024):** 20 seed examples and 1,180 disjoint evaluation examples under the standard four-option setting.

Leakage control. Seed and evaluation sets are disjoint at the underlying problem level, including across aligned MathVerse variants: no version of an evaluation problem is used for seed construction. Each MathVerse version uses an independently constructed library, frozen before evaluation. Evaluation examples are excluded from all library construction and refinement.

Evolution protocol. The main setting performs 20 offline evolution rounds on the 20-example seed set while maintaining a 20-skill library. The first round converts admitted seed trajectories into initial skill profiles; subsequent rounds refine them through selective evolution and extraction using seed experience and observation, reasoning, and verification feedback. Reflection retries repair individual solve attempts and are separate from evolution rounds. Full MUSE uses the library after round 20, whereas **w/o Evol.** retains the first-round profiles.

A.2 Baseline controls

All compared methods use the same backbone and evaluation examples. Direct receives only the task input with tool-call support. MUSE retrieves candidate profiles from its fixed 20-skill library and

performs skill-guided solving and reuse gating as shown in Figure 2. Skill construction and refinement remain offline. Seed-derived skills and memories use the same 20 labeled randomly sampled seed examples.

The w/o Evol. ablation uses the multimodal skills summarized in the first round and disables subsequent skill evolution. On GPT-5.4/MATHVERSE, it achieves 79.0%, compared with 62.1% for Direct and 93.4% for full MUSE. The 16.9-point difference from Direct reflects the benefit of using first-round multimodal skills, while the additional 14.4-point improvement compares full evolution with retaining only those initial skills. The first-round and fully evolved skill descriptions have similar lengths, so this comparison also controls approximately for the amount of skill text. These contrasts distinguish first-round skill summarization from subsequent skill evolution.

Self-Consistency and inference budgets. Self-Consistency runs $k = 5$ independent Direct-style rollouts on the same task and image, without skills, memories, or evaluation labels; normalized final answers are combined by majority voting. Reflexion instead adds one critique-and-revise step to an initial Direct answer. MUSE performs retrieval, guided solving, and reuse gating with a frozen library. These are distinct inference protocols: five rollouts do not necessarily imply five model calls, and the shared backbone and evaluation split do not establish equal total inference cost. Offline reflection retries belong to library construction and are not an online retry budget. The current report does not specify Self-Consistency tie-breaking or per-method model-call and token totals.

A.3 Evolution Quality Gate vs. Memory Retrieval

Key Observation. AMem performs at or below Direct on three of four benchmarks, with MathVerse showing a striking -3.8pp drop. This indicates that *retrieval without evolution is insufficient or even harmful*: unfiltered memory can introduce mismatched procedures. The large AMem-to-MUSE gains (+35pp MathVerse, +17pp MMMU-Pro) support selective admission and skill abstraction as a combined evolution mechanism over raw retrieval; this comparison does not isolate their individual contributions.

Setting	Value / scope
Seed examples	20 per constructed library
Evolution rounds	20 offline rounds
Library size	20 profiles
Reflection retries	At most 2 per offline attempt
Self-Consistency	$k = 5$ independent rollouts
Reflexion	One critique-and-revise step
Retrieval threshold	$\tau = 4.5$
Reuse threshold	0.64 if support ≥ 2 ; 0.92 otherwise
Threshold selection	Separate 20-example M3CoT validation subset
Evaluation state	Skill contents and trust frozen
Reported results	Mean over 3 library reconstructions

Table 2: Explicit settings reported for the evaluated protocol. Retrieval top- K and Self-Consistency rollout count k denote different quantities.

Table 3: MUSE’s gains over raw memory retrieval. AMem retrieves raw trajectories without selective admission, while MUSE admits only verified, abstracted skills. “MUSE - AMem (pp)” reports the accuracy difference in percentage points between MUSE and the evaluated AMem baseline; it does not isolate the effect of evolution.

Benchmark	Direct	AMem	MUSE	MUSE - AMem (pp)
MathVista	78.98	79.29 (+0.31)	83.11	+3.82pp
MathVerse	62.10	58.30 (-3.80)	93.40	+35.10pp
MMMU-Pro	63.90	63.81 (-0.09)	81.10	+17.29pp
M3CoT	85.08	84.42 (-0.66)	91.42	+7.00pp
Average	72.52	71.46 (-1.06)	87.26	+15.80pp

A.4 Evolution-Stage Skill Construction & Evaluation

Skill construction and refinement are offline. The main setting uses 20 offline evolution rounds on 20 randomly sampled labeled seed examples and maintains a library of 20 skills. Full MUSE uses the round-20 library; w/o Evol. retains the first-round library. The library is independently rebuilt in each of the three runs and frozen before evaluation. At inference, the retrieval agent and skill solver agent use the frozen library for retrieval, guided solving, and reuse gating.

Skills are constructed only from seed experience and frozen before evaluation. Evaluation labels are used only for scoring after predictions; evaluation instances cannot change skill contents or trust state.

The main setting contains 20 skills per separately constructed set, including each MathVerse version.

Round-level inputs and outputs. Let $\mathcal{S}^{(t)}$ denote the library after evolution round t , and let $\mathcal{T}_{\text{adm}}^{(t)}$ denote the seed trajectories admitted under

Equation 1 in that round. Each admitted trajectory supplies the task description without the gold answer, observations o_i , reasoning summary r_i , verifier feedback v_i , and final prediction $\hat{y}_i$. The first round abstracts this experience into $\mathcal{S}^{(1)}$. Later rounds refine the current library using admitted seed experience and its observation, reasoning, and verification feedback. At the library level, the data flow can be written as

$$\begin{aligned}\mathcal{S}^{(1)} &= \text{Extract}(\mathcal{T}_{\text{adm}}^{(1)}), \\ \mathcal{S}^{(t)} &= \text{Refine}(\mathcal{S}^{(t-1)}, \mathcal{T}_{\text{adm}}^{(t)}), \quad 2 \leq t \leq 20.\end{aligned}\tag{5}$$

Here, Extract invokes the skill extraction template without an existing profile, while Refine invokes that same template with the current profile and admitted independent seed experience. They describe initial summarization and subsequent library refinement through the selective-evolution and extraction process in Section 3; they are not additional solver or verifier modules. The refinement output remains a library of profiles with the same five fields $(\phi_i, h_i^v, h_i^r, h_i^z, k_i)$. Thus, iteration changes the procedural guidance available for later tasks, while preserving the skill representation and the 20-skill library size. Full MUSE deploys $\mathcal{S}^{(20)}$; w/o Evol. deploys $\mathcal{S}^{(1)}$.

Round execution and writing boundary. For each seed task, the observation pass produces o_i , the reasoning pass produces $(r_i, \hat{y}_i)$, and the verifier supplies v_i . A verifier rejection can trigger at most two reflection retries; follow-up visual questions are returned to the observation pass. After the final attempt, Equation 1 is applied outside the prompts. An admitted independent trace is passed to the extraction template, together with the current profile when refining; a rejected trace supplies no new skill-writing evidence. The writer returns one complete profile in the same JSON schema. Its field changes express corrections to observation, reasoning, applicability, or verification. These updates modify the skill profile without introducing additional rule-engine operations. A skill-guided solve is checked under Equation 3 for offline reuse support, and cannot be relabeled as independent experience. Gold answers are not supplied to the writer. Reflection retries update a solve attempt, whereas extraction or refinement updates a skill profile.

Trajectory admission and reuse qualification. The two gates apply to different objects. Equa-

tion 1 determines whether a seed solve trace can supply experience for skill extraction: the solve must be independent of existing skills, free of runtime errors, and answer-correct. Equation 3 instead qualifies skill reuse from the checking decision, answer canonicalization, and confidence. During construction, accepted reuse decisions contribute to the trust state; at evaluation, gating uses the frozen trust state. Neither operation turns a skill-guided trace into an independently solved trajectory or overrides the admission gate. Gold answers are used only for the seed admission comparison, whereas the skill-writing input contains the solver prediction and its supporting experience. Reflection retries repair a solve attempt before admission; an evolution round updates the library. These are separate levels of iteration.

Online retrieval and reuse gating. The retrieval agent applies the scored interface $C(x_j)$ in Equation 4 to the frozen library. It supplies retrieved profiles to the skill solver agent, which checks applicability and uses their visual, reasoning, and verifier hints. Reuse Gating applies Equation 3 to the skill-guided result. These are the inference components shown in Figure 2 and described in Section 3.3.

Deployment boundary. The final library and trust state are frozen before any evaluation instance is solved. Evaluation supplies neither refinement experience nor trust updates. The online output is the solver’s prediction, followed by answer extraction and normalization for scoring. The construction templates in Appendix C document seed solving and extraction, while the online retrieval and reuse-gating description documents deployment.

A.5 Hyperparameters Selection

Retrieval threshold. The retrieval threshold $\tau=4.5$ is selected on a held-out validation subset of M3CoT (20 examples disjoint from both evolution and evaluation splits) and then fixed across all benchmarks. No evaluation-set labels are used for threshold tuning. At evaluation, the retrieval agent applies this fixed threshold to candidate selection as described in Section 3.3.

Trust-dependent confidence threshold. The confidence threshold $\theta(\text{trust}_i)$ uses an explicit two-stage scheme: skills with ≥ 2 previous offline reuse decisions accepted by the label-free gate in Equation 3 use $\theta=0.64$; lower-support skills use $\theta=0.92$.

These counts measure gate acceptance, not gold-label correctness. The thresholds are set on the same M3CoT validation subset.

A.6 Ablation Condition Definitions

The w/o Evol. condition retains the multimodal skills produced by the first round of skill summarization and disables all subsequent skill evolution. This condition uses profiles generated during the initial summarization round. In contrast, w/o MM retains text skills but removes multimodal skill information. The two conditions therefore test different ingredients: evolution beyond first-round summarization and multimodal skill content, respectively. The w/o multimodal skill condition retains text skills for retrieval and reuse but removes multimodal descriptions from the skill profiles. This means the model can still access the image, but it cannot utilize the multimodal information in its skill-processing process. This is not a no-skill baseline. The detailed removal prompt for this ablation is:

Ablation: Remove all multimodal information from the skill profile.

You are given a multimodal skill profile that contains visual hints, reasoning hints, verifier hints, and retrieval descriptors. You are asked to remove all visual hints and any other multimodal information from the skill profile while preserving the reasoning hints, verifier hints, and retrieval descriptors. The resulting skill profile should be text-only and should not contain any references to visual information or multimodal content.

A.7 Reuse diagnostics.

To make skill reuse auditable, we log a per-instance diagnostic trace.

Retrieval and reuse trace. The diagnostic trace covers retrieved candidates, retrieval scores, selected profiles, and reuse-gating outcomes.

Reuse qualification. Reuse Gating uses the qualification rule in Equation 3 to check skill-guided results without evaluation labels.

Trust state. Trust is accumulated during offline evolution and frozen with the skills before evaluation. Inference uses retrieval, skill-guided solving, and reuse gating with this frozen state.

Metric scope. Offline reuse frequency is the fraction of offline reuse opportunities accepted by Equation 3. Evaluation reuse accuracy is the proportion of correct final answers within the evaluation skill-use subset; overall accuracy uses all evaluation instances. Online coverage would be the fraction of evaluation instances accepting a skill result. It is distinct from offline reuse frequency

Table 4: Skill reuse diagnostics per backbone and benchmark. Here, reuse frequency is the skill reuse rate during evolution; reuse accuracy is final-answer accuracy within the skill-use subset. *Reuse%*: reuse frequency. *Reuse Acc.*: reuse accuracy.

Backbone	Benchmark	Reuse%	Reuse Acc.
GPT-5.4	MathVista	31.3%	93.8%
GPT-5.4	M3CoT	99.8%	91.6%
GPT-5.4	MathVerse	100.0%	93.4%
GPT-5.4	MMMUI-Pro	99.2%	81.6%
GPT-5.4-mini	MathVista	24.9%	93.0%
GPT-5.4-mini	M3CoT	97.7%	87.7%
GPT-5.4-mini	MathVerse	99.5%	86.9%
GPT-5.4-mini	MMMUI-Pro	97.7%	73.5%
Gemini-3-Flash	MathVista	21.0%	97.9%
Gemini-3-Flash	M3CoT	80.0%	98.9%
Gemini-3-Flash	MathVerse	60.7%	100.0%
Gemini-3-Flash	MMMUI-Pro	53.9%	96.7%
Claude-Sonnet-4.6	MathVista	40.1%	98.6%
Claude-Sonnet-4.6	M3CoT	44.8%	95.0%
Claude-Sonnet-4.6	MathVerse	92.4%	94.6%
Claude-Sonnet-4.6	MMMUI-Pro	93.5%	95.0%
GLM-4.6-V	MathVista	29.8%	95.3%
GLM-4.6-V	M3CoT	80.8%	88.1%
GLM-4.6-V	MathVerse	73.4%	92.1%
GLM-4.6-V	MMMUI-Pro	67.8%	76.2%

and is not inferred from the reported offline percentages. Trajectory admission under Equation 1 is a separate statistic and is not called reuse frequency.

Outcomes and summaries. Here, reuse frequency is the skill reuse rate during evolution; reuse accuracy is final-answer accuracy within the skill-use subset. *Reuse%* and *Reuse Acc.* denote reuse frequency and reuse accuracy, respectively. Table 4 reports these diagnostics.

A high reuse frequency does not imply a large skill set: the same skill can guide many instances. GPT-5.4 has skill reuse rates of 99.8%, 100.0%, and 99.2% on M3CoT, MathVerse, and MMMUI-Pro, respectively. The main setting uses 20 fixed skills.

B Split and Leakage Diagnostics

These diagnostics test two alternative explanations for the main gains. First, the primary randomly sampled seed split might be unusually favorable. Table 5 therefore reruns GPT-5.4-mini with three additional random 20-example seed splits while keeping the same frozen-library evaluation protocol. Results remain within 1.70 percentage points of the primary split on every benchmark, supporting robustness to the choice of seed examples.

Table 5: Multi-split robustness check for GPT-5.4-mini. The primary split and three additional random splits each use 20 labeled randomly sampled seed examples and the same frozen-library evaluation protocol.

Benchmark	Primary split	Random-1	Random-2	Random-3
MathVista	73.98	73.10	74.35	72.85
M3CoT	86.83	85.90	87.25	86.10
MathVerse	86.50	84.80	85.70	86.20
MMMU-Pro	72.20	70.95	72.85	71.60

Table 6: Order and leakage-control diagnostics on GPT-5.4-mini. Controls preserve the same solver and inference budget while perturbing only the skill library or retrieval signal.

Condition	MathVista	M3CoT	MathVerse	MMMU-Pro
Direct	71.94	77.83	58.20	52.29
MUSE primary split	73.98	86.83	86.50	72.20
MUSE additional random split	73.10	85.90	84.80	70.95
Shuffled skill assignment	70.85	76.40	59.10	51.70
Wrong-domain skill library	71.30	77.20	60.45	52.10
Metadata-only retrieval	72.05	80.30	67.90	56.20

Second, retrieved skills might reproduce information from previously solved examples. Table 6 perturbs only the library or retrieval signal while preserving the solver and inference budget. Shuffling skill assignments and using wrong-domain libraries collapse performance back toward Direct, while metadata-only retrieval keeps only a smaller fraction of the gain. These controls support the importance of matching the retrieved visual-reasoning guidance to the current task while holding the solver and inference budget fixed. Skill reuse follows the retrieval and gating process in Figure 2.

C Prompt Templates

The observation, reasoning, verifier, extraction/refinement, and retry templates below document offline evolution. Extraction and refinement share a schema; refinement additionally receives the current profile. At evaluation, the retrieval agent supplies frozen skill profiles to the skill solver agent for guided solving and reuse gating.

Multimodal Reasoning Solver: observation pass.

This prompt instantiates the structured visual-observation part of the *Multimodal Reasoning Solver* in Section 3.2.1. It produces the observation component o_i of the trajectory by extracting image-grounded evidence only; it is explicitly forbidden from solving the task. The observation stage records explicit visual evidence before answer generation.

Multimodal Reasoning Solver: observation pass

You are the multimodal_reasoning_solver running its observation pass for a multimodal reasoning pipeline. Look at the image and extract only grounded visual evidence. Return JSON only with keys: scene_type, observation_strategy, visual_facts, uncertainties, focus_answers. Rules: select an observation strategy appropriate to the visual content, such as OCR extraction, object detection, diagram parsing, chart-axis reading, or spatial-relation analysis. Prefer explicit visible text, numbers, labels, flags, logos, jersey names/numbers, podium seals, event branding, charts, and object counts. For public figures, do not rely on face recognition alone; mention identity hypotheses only when supported by visible text or strong non-facial contextual clues in the image. For age questions, provide only coarse age ranges with clear uncertainty if exact ages are not visible. If a focus question cannot be answered from the image, say so explicitly. Do not solve the problem or provide a final answer/choice in the visual stage. The answer field in focus_answers must be null. Put only visible observations in visual_facts/supporting_clues; do not perform algebra, geometry solving, or pattern completion. Each focus_answers item should contain: question, answer, confidence, evidence_type, visual_observations, supporting_clues, person_age_ranges, relative_order.

Multimodal Reasoning Solver: reasoning pass.

This prompt implements the reasoning part of the *Multimodal Reasoning Solver*. It consumes the task and structured observations o_i , produces reasoning summary r_i and candidate prediction $\hat{y}_i$, and can request additional visual checks when the evidence is insufficient. Its output fields correspond to the trajectory fields used by the skill evolver.

Multimodal Reasoning Solver: reasoning pass

You are the multimodal_reasoning_solver running its reasoning pass for a multimodal QA pipeline. You receive a task, structured visual evidence, optional answer choices, and optionally retrieved skill profiles. Return JSON only with keys: reasoning_steps, candidate_answer, answer_confidence, needs_visual_recheck, focus_questions, normalization_notes. Be concise and canonical. Treat retrieved skills only as procedural guidance about what to inspect, how to reason, and how to verify; never copy an evolution answer or use a skill unless it matches the current image. For integer/float questions, candidate_answer should be a scalar or null. If evidence is insufficient, set candidate_answer=null and ask targeted focus_questions. Treat focus_answers from the observation pass as observations only, not as proposed answers. If your reasoning derives a value or option, candidate_answer must exactly match that derivation.

Multimodal Reasoning Verifier.

This prompt implements the *Multimodal Reasoning Verifier* described in Section 3.2.1 during offline evolution. The construction verifier checks seed trajectories; Equation 3 also defines the inference-stage reuse gate. It independently re-examines the current image and checks grounding, internal consistency,

1025 and answer format. Canonicalization itself is per-
1026 formed by the scorer outside the prompt.

1027

Multimodal Reasoning Verifier

You are the multimodal_reasoning_verifier. You receive the current image, task, structured visual evidence, reasoning steps, optional retrieved skill guidance, and a candidate answer. Independently re-examine the current image and decide whether the candidate answer is grounded enough to accept. Return JSON only with keys: decision, issues, revised_answer, follow_up_visual_questions, confidence. The decision must be one of accept, reject, or recheck. Rules: accept only when the candidate answer is consistent with the visible evidence, the reasoning steps, and the expected answer format. If reasoning_steps derive one value/choice but candidate_answer is different, set decision='reject' and put the derived value in revised_answer for a solver retry; do not accept it directly. If evidence is insufficient, set decision='recheck' with targeted follow_up_visual_questions. Do not accept answers justified by contradictory language such as 'therefore X' followed by an unsupported 'however Y'.

1028

1029

1030

1031

1032

1033

1034

1035

1036

1037

1038

1039

Multimodal Skill Extraction Agent. This prompt implements the *Multimodal Skill Extraction Agent* in Section 3. It receives the admitted evolution trajectory fields described there and, for refinement, the current profile, including the task description without the gold answer, visual observations, reasoning summary, verifier feedback, and final prediction, and distills a reusable profile with applicability, visual, reasoning, verifier, and retrieval fields. The prompt prohibits instance-specific answer copying: the skill guides reasoning over the current image without storing solved examples.

Multimodal Skill Extraction Agent

You are the multimodal_skill_extraction_agent, a reusable skill-profile writer for a multimodal reasoning system. You will read an admitted solve trace containing the task description without the gold answer, visual observations, reasoning summary, verifier feedback, and final prediction. You may also receive a current skill profile. If no current profile is supplied, produce one initial reusable multimodal skill profile. Otherwise, return one revised profile in the same schema: retain useful guidance, revise fields supported by the admitted trace and its verifier feedback, and incorporate transferable corrective checks from successful reflection retries. Do not use a skill-guided or rejected trace as independent writing evidence. Do not add task conditions or checks unsupported by the provided experience. Your goal is to summarize transferable experience, not to memorize the sample. Return JSON only with keys: applicability, visual_hint, reasoning_hint, verifier_hint, keywords, retrieval_text_bge, retrieval_text_clip, confidence. The applicability, visual, reasoning, and verifier hints should each be one to two concise sentences. Describe the reusable procedure and incorporate transferable corrective checks from reflection into the same skill text. Do not include the sample's final answer verbatim. Do not memorize sample-specific names, dates, exact measurements, or unique labels unless they are clearly reusable category cues. Prefer category-level strategies such as when the skill applies, what visual evidence matters, how to structure the reasoning, and how to verify the final answer form.

1040

Multimodal Reasoning Solver: reflection retry pass. This prompt corresponds to the bounded evolution-stage retry described under selective evolution: when an evolution attempt fails without a runtime error, the solver receives the verifier's rejection reason and re-attempts reasoning. The same retry-pass template can be invoked for up to two turns, matching the Method section. It does not create a saved skill directly; only if the retried trajectory passes the admission gate can its transferable corrective guidance be incorporated into a skill.

1041

1042

1043

1044

1045

1046

1047

1048

1049

1050

1051

1052

Multimodal Reasoning Solver: reflection retry pass

You are the multimodal_reasoning_solver running an evolution-stage reflection retry. You receive a failed evolution-stage trace and the verifier's rejection reason, then re-attempt the solve. Return JSON only with keys: retry_reasoning_steps, candidate_answer, answer_confidence, needs_visual_recheck, focus_questions, normalization_notes. Use the verifier's rejection reason to decide what visual evidence to re-check and what reasoning error to avoid. Do not create or save a skill. Do not use or infer the gold answer. Do not memorize sample-specific names, dates, exact measurements, or unique labels unless they are visible evidence needed for the current solve.

1053

Online skill retrieval and guidance. At evaluation, the retrieval agent uses the current task to retrieve candidate profiles from the frozen library. The skill solver agent checks applicability against

1054

1055

1056

1057

Table 7: M3CoT accuracy (%) for open-source LVLMS.

Method	InternVL	Qwen3-VL
Direct	54.42	59.92
Self-Consistency	54.00	65.08
Reflexion	45.83	54.33
AMem	52.33	61.25
MUSE	65.17	67.92

the current image and question, uses the retrieved visual, reasoning, and verifier hints, and performs reuse gating under Equation 3. Answer extraction and normalization follow the resulting prediction, as shown in Figure 2.

Self-Consistency aggregation. The baseline uses the Direct task input for each of its $k = 5$ independent rollouts. Answer extraction and normalization precede majority voting, so equivalent answer forms count together. The aggregation is external to the solver prompt; it introduces no skill-extraction, refinement, or reuse-gating prompt.

Direct evaluation input. Direct receives the task with tool-call support. MUSE uses retrieved frozen skills through the inference process in Figure 2. Neither receives evaluation labels.

Reflection baseline. This prompt implements the Reflexion baseline in Section 4.1. It critiques and revises only the Direct answer, with no access to the evolution skill library, AMem memories, generated profiles, or gold labels. Because the baseline has no access to the skill library, this comparison isolates self-reflection.

Reflection baseline

You are a reflection baseline for a multimodal QA model. You receive the original image, task, answer choices when available, and the model’s initial direct answer. Critique only that answer using the current image, then either keep it or revise it. Do not use external memory, generated skills, tools, or hidden gold labels. Return JSON only with keys: critique, revised_answer, confidence.

D Open-Source Model Analysis

Table 7 reports the M3CoT results corresponding to Figure 3. MUSE improves InternVL3.5-8B by 10.75% and Qwen3-VL-8B-Thinking by 8.00% over Direct, achieving the highest accuracy among the compared methods for both models. This focused evaluation extends the evidence for skill reuse to open-source 8B backbones.

Table 8: Accuracy (%) at the Science and Mathematics stages. Each row evaluates the domain of the current evolution stage. Freeze denotes the skill library frozen after Commonsense evolution.

Evaluation domain	No Evolution	Freeze	MUSE
Science	76.4	80.2	82.5
Mathematics	73.7	76.6	79.7

E Additional Analysis and Discussion

Cross-domain stage comparison. Figure 6(a) evaluates Commonsense, Science, and Mathematics at their respective stages of the evolution sequence. The evaluation domain therefore changes with the stage. Continual MUSE scores 84.5%, 82.5%, and 79.7%, respectively. Table 8 summarizes the same stage-wise comparison. At the Science and Mathematics stages, continual evolution improves over both No Evolution and the library frozen after Commonsense. Between-stage score changes also reflect differences in domain difficulty and do not measure forgetting on a fixed earlier-domain test set. Figure 6(b) separately varies evolution size.

Construction and inference cost. Skill construction is an offline operation with an average reported cost of \$32.6 on GPT-5.4. At evaluation, the frozen library supports retrieval, skill-guided solving, and reuse gating. These inference operations are distinct from offline construction; their overhead cannot be inferred from the construction cost or the number of stored skills. On GPT-5.4 and Math-Vista, overall accuracy is 83.11%, compared with 78.98% for Direct; reuse accuracy is 93.8%, the final-answer accuracy within the skill-use subset.

Repeated construction. Each of the three runs independently reconstructs the skill library before evaluation; the main table reports mean accuracy across these runs. The reported run-to-run variation is approximately $\pm 3\%$. This aggregate variation is not a per-cell standard deviation or confidence interval.