πSKILL: STEP-AWARE EXPERIENCE DISTILLATION AND RELIABLE MEMORY REUSE FOR MULTIMODAL AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal agents can reuse past interactions through external memory, yet trajectory-derived memories are often both coarse and unreliable: a rollout-level summary can miss locally useful decisions, while unverified experiences may remain in the prompt after they become irrelevant or repeatedly harmful. We introduce πSKILL, a training-free framework that separates experience extraction, maintenance, and use. First, Circular Step-level Knowledge Distillation (CSD) combines cross-rollout circular analysis of useful operations and observed failure points to identify operational lessons at different granularities. A deterministic admission rule limits overly specific, redundant, and non-actionable candidates. Second, Confidence-aware Memory Updating (CMU) associates each admitted experience with outcome statistics and a lifecycle state, allowing unreliable entries to be suppressed as feedback accumulates. At inference time, the agent decomposes the task into method-oriented retrieval queries and rewrites a small set of selected experiences into task-specific guidance. Using Qwen2.5-VL-7B-Instruct, πSKILL improves the four-benchmark macro-average Average@4 from 12.67 to 15.32 and Pass@4 from 25.28 to 33.33 relative to state of the art. The gains are largest on benchmarks that require repeated procedural decisions, while the ablations show that inference-time contextualization is essential and that the two memory-construction components have task-dependent effects.

1 INTRODUCTION

Multimodal large language models have enabled agents to interleave visual reasoning with external tools, including code execution, image transformation, and web search Wang et al. (2025; 2024a); Guo et al. (2025); Chu et al. (2026). Such agents can address tasks that are difficult to solve through passive perception alone, but their failures are often recurrent. An agent may repeatedly choose an unsuitable crop, apply optical character recognition before correcting image orientation, or retrieve a procedure that was useful for a superficially similar but operationally different task. Updating model parameters after every interaction is expensive and can be undesirable when tasks or tools change quickly. Geng et al. (2025); Liu et al. (2025) External experience memory offers a lighter-weight alternative: the agent records lessons from completed trajectories and retrieves them for future tasks Shinn et al. (2023); Zhao et al. (2024); Wang et al. (2024b).

The central difficulty is not simply storing more trajectories. A useful memory system must decide *what to extract*, *which experiences remain trustworthy*, and *how a general lesson should influence the current task* Ouyang et al. (2025). Existing systems commonly summarize a complete rollout into one memory item Suzgun et al. (2026); Jiang et al. (2026); Luo et al. (2026a). This captures the overall strategy but can obscure local decisions that determine success or failure. Wang et al. (2026a;b) Conversely, extracting many local lessons increases coverage at the cost of redundancy and noise Zhang et al. (2026). Memory quality also changes with use: an initially plausible rule may repeatedly fail, become obsolete, or conflict with a better-supported alternative Wu et al. (2025); Xiong et al. (2026); Lam et al. (2026); Chao et al. (2026); Yadav (2026); He et al. (2026) . Finally, semantic similarity alone does not guarantee that a retrieved memory is operationally relevant to the agent's current stage of reasoning Luo et al. (2026b); Huang et al. (2026).

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.

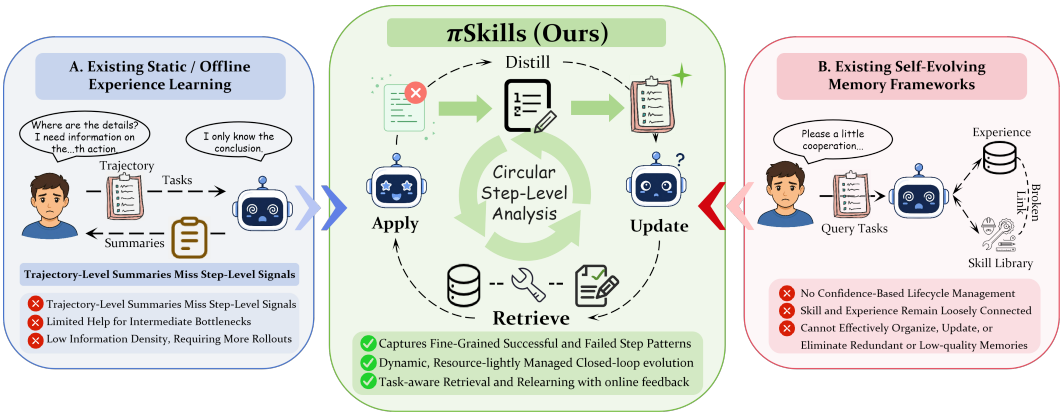


Figure 1: Experience-learning paradigms. Trajectory-level memory provides global guidance but can hide local causes of success and failure. πSKILL combines cross-rollout evidence with quality-gated local lessons and maintains outcome-conditioned metadata for subsequent retrieval.

We study whether these three decisions can be separated within a training-free multimodal agent. πSKILL uses (i) multi-granularity experience distillation, (ii) outcome-conditioned memory maintenance, and (iii) task-conditioned retrieval and rewriting. During memory construction, cross-rollout comparison identifies strategies that distinguish successful and unsuccessful attempts, while single-rollout analysis proposes local operational lessons. A transparent admission rule limits candidates that are overly specific, redundant, or difficult to execute. Each admitted experience receives reliability and lifecycle metadata that are updated from subsequent task outcomes. At test time, method-oriented subqueries retrieve a small, diverse set of active experiences, which are rewritten against the current query and visual input before being added to the agent context. Figure 2 summarizes the pipeline.

We evaluate πSKILL on VisualToolBench Guo et al. (2025), TIR-Bench Li et al. (2025a), MMSearch-Plus Tao et al. (2025), and AgentVista Su et al. (2026) using a shared Qwen2.5-VL-7B-Instruct backbone and four rollouts per task. πSKILL improves the macro-average Average@4 by 2.65 absolute points and Pass@4 by 8.05 points over XSkill. The gains over XSkill are largest on VisualToolBench and AgentVista. Component analyses show that the complete pipeline improves the macro-average, while the effects of CSD and CMU vary across benchmarks.

Our contributions are:

- We formulate experience reuse as three separable decisions—extraction, maintenance, and contextualized use—and instantiate them in a training-free multimodal-agent framework.
- We introduce CSD, which combines cross-rollout comparison with quality-gated single-rollout analysis to improve the granularity of operational experience while controlling memory growth. Then we introduce CMU and a decomposition-aware inference pipeline that jointly account for experience reliability, retrieval diversity, and task-conditioned rewriting.
- We evaluate the framework across four heterogeneous benchmarks, additional backbones, a cross-task transfer setting, and controlled component and hyperparameter analyses.

2 Related Work

Multimodal tool-using agents. Multimodal agents combine visual understanding with external tools such as code execution, image processing, and web browsing Li et al. (2025b). Recent benchmarks test tool-enabled image reasoning, thinking with images, multimodal search, and realistic long-horizon interaction Guo et al. (2025); Li et al. (2025a); Su et al. (2026). Although stronger perception and tool interfaces expand the range of solvable tasks Hong et al. (2025), many systems still treat each task as an isolated episode. Intermediate decisions, tool-use patterns, and execution feedback are therefore discarded after the final answer Li et al. (2025c).Recent work on multimodal long-term memory further highlights the challenges of retaining and organizing information across

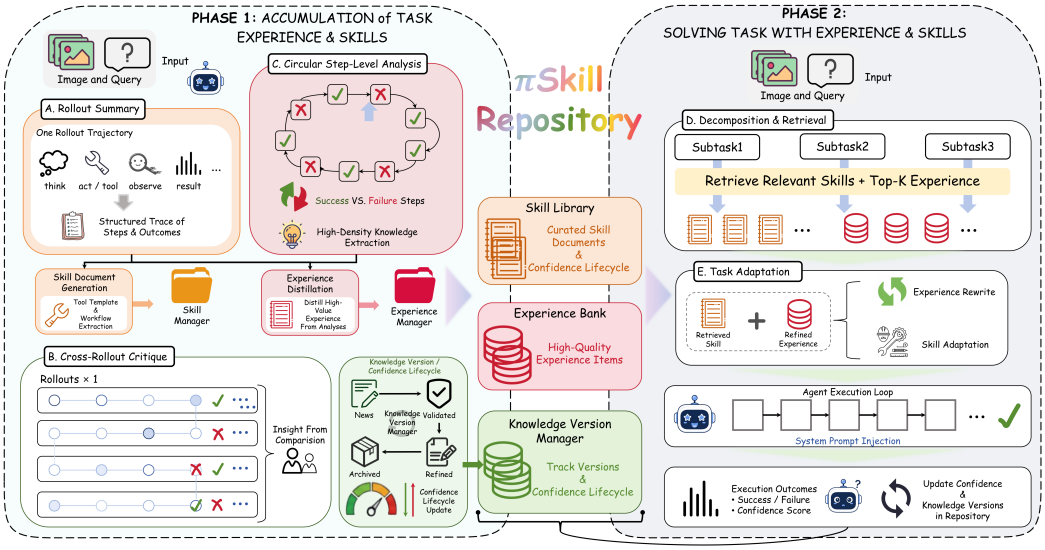


Figure 2: Overview of πSKILL. Phase 1 follows panels A–C: rollout summarization and skill generation, cross-rollout critique, and circular step-level analysis populate the skill library and experience bank. The knowledge-version manager maintains confidence and lifecycle metadata. Phase 2 follows panels D–E: decomposition and retrieval select relevant skills and experiences, task adaptation rewrites them for the current input, and execution outcomes update the repository.

extended interactions **?**.πSKILL instead focuses on retaining operational knowledge that can guide later multimodal interactions.

Experience and memory learning. Language agents can improve without parameter updates by converting previous attempts into verbal feedback, workflows, or reusable procedures Shinn et al. (2023); Zhao et al. (2024); Wang et al. (2024b). Dynamic Cheatsheet maintains an adaptive test-time memory Suzgun et al. (2026), Agent-KB organizes cross-domain agent knowledge Tang et al. (2025), and ReasoningBank extracts reusable reasoning strategies Ouyang et al. (2025). A-MEM further organizes memories through dynamic indexing and linking, allowing existing representations to evolve as new knowledge is incorporated Xu et al. (2026). In multimodal settings, XSkill combines trajectory-derived experiences with task-level skills Jiang et al. (2026). Related work has also emphasized that memory growth, staleness, and following behavior can determine whether retrieval helps or harms an agent Hu et al. (2025); Wu et al. (2025); Xiong et al. (2026). πSKILL builds on this line of work but separates three issues that are often conflated: the granularity of extracted experience, the reliability of stored memory, and the contextualization of retrieved guidance.

3 METHOD

3.1 OVERVIEW: A TWO-PHASE WORKFLOW

Let a task be $x = (q, I, \mathcal{U})$, where q is the user query, I is the visual input, and $\mathcal{U}$ is the available tool set. The agent produces N rollouts for an accumulation task. Rollout i is

$$\tau_i = \{(a_{i,t}, o_{i,t})\}_{t=1}^{T_i}, \tag{1}$$

where $a_{i,t}$ contains the assistant decision and tool invocation at step t, and $o_{i,t}$ is the corresponding textual or multimodal observation. Each rollout receives an outcome $y_i \in \{0, 1\}$ from the benchmark evaluator. The objective is to convert the rollout set into an experience memory $\mathcal{M}$ and use a small subset of that memory to improve future tasks without updating model parameters. πSKILL follows the two phases and five labeled stages in Figure 2. During **Phase 1: accumulation of task experience and skills**, Stage A converts raw trajectories into structured rollout summaries and task-level skill documents; Stages B and C extract cross-rollout and step-level experiences; and the central repository

stores the resulting skills, experiences, and lifecycle records. During **Phase 2: solving a task with experience and skills**, Stage D decomposes the incoming task and retrieves a sparse memory subset, while Stage E adapts the retrieved knowledge before prompt injection. The resulting execution outcome closes the feedback loop by updating confidence and lifecycle metadata.

3.2 Stage A: Rollout Summarization and Skill Generation

For each accumulation task, the agent first converts every raw trajectory into a structured sequence of actions, tool calls, observations, errors, and the final task outcome. The rollout summary preserves the order of interaction and explicitly records which tools were invoked, what evidence they returned, and where the trajectory succeeded, backtracked, or failed. These summaries provide a common representation for the two experience-extraction paths in Stages B and C.

In parallel, a skill generator abstracts the trajectory into a compact task-level workflow. Unlike an experience item, which encodes one conditional operational lesson, a skill document organizes a reusable multi-step procedure. Skill documents are periodically merged and pruned before entering the skill library. This distinction matches the central repository in Figure 2: the skill library stores curated procedures, whereas the experience bank stores localized decision rules.

3.3 Stages B–C: Circular Step-Level Knowledge Distillation

Stage B: cross-rollout critique. When multiple rollouts are available for the same task, CSD contrasts successful and unsuccessful attempts. The critic receives the task, the ordered trajectories, and their outcomes, and proposes a small set of *core experiences* that capture decisions associated with different outcomes. This branch retains the global comparison signal used by XSkill while making the evidence source explicit.

Stage C: circular step-level analysis. Cross-rollout comparison can miss useful local behavior when all attempts share the same outcome or when only one trajectory contains an informative tool interaction. We therefore analyze each rollout independently:

$$E_i^{\text{local}} = C_\theta(x, \tau_i, y_i), \tag{2}$$

where C_θ is an LLM critic. The critic revisits the ordered trajectory to contrast apparently useful operations with observed failure points. It first scans forward to identify operations that produced useful observations, then uses the final outcome and recorded errors to revisit failure points and formulate corrective rules. The term *circular* refers to this return from the trajectory outcome to the local step sequence and to the later feedback from Phase 2; it does not imply gradient-based optimization. The critic proposes positive lessons from useful operations and cautionary lessons from failures. We use *step-level* rather than *step-supervised*: the critic has access to step boundaries, but no external causal label is assumed for every transition.

Quality-gated admission. Single-rollout critique increases recall but can produce generic or sample-specific rules. For a candidate e, we compute

$$q(e) = \text{clip}\left(q_0 + w_c C(e) + w_a A(e) - w_s S(e) - w_v V(e), 0, 1\right), \tag{3}$$

where $C(e)$ indicates an explicit trigger condition, $A(e)$ indicates an executable action, $S(e)$ detects sample-specific content, and $V(e)$ detects vague advice. Candidates with $q(e) < \eta$ are rejected. We then remove near-duplicates of higher-ranked core experiences. At most one local experience is retained per rollout group in the default setting, and local experiences occupy at most a fraction ρ of each batch, subject to an absolute cap. Similar experiences are periodically consolidated to control library growth. Appendix A reports the exact scoring rule and thresholds.

3.4 Repository Construction and Confidence-Aware Memory Updating

As shown at the center of Figure 2, the repository contains three coordinated stores. The **skill library** contains curated workflow documents. The **experience bank** contains quality-gated operational items. The **knowledge-version manager** stores confidence, use statistics, lifecycle state, and version history. This separation allows the system to revise reliability metadata without rewriting the semantic content of every memory item.

For each textual experience e_j, CMU maintains

$$m_j = (c_j, n_j^+, n_j^-, z_j, v_j, t_j, R_j), \tag{4}$$

where c_j is confidence, n_j^+ and n_j^- count successful and failed attributed uses, z_j is the lifecycle state, v_j is the version index, t_j is the last-use time, and R_j stores links to related skills. New experiences are initialized with confidence c_{init} and state `active`.

Outcome-conditioned update. When a retrieved experience can be traced to a completed execution, its confidence is updated from the execution outcome:

$$c_j \leftarrow \begin{cases} \min(1, c_j + \Delta_+), & \text{if the execution succeeds,} \\ \max(0, c_j - \Delta_-), & \text{otherwise,} \end{cases} \tag{5}$$

with $\Delta_- > \Delta_+ > 0$. A larger penalty reflects the asymmetric risk of repeatedly injecting harmful guidance. If failures exceed both a fixed review threshold and the number of successful uses, the experience enters `to_review`. Version snapshots preserve the update history for inspection. Because task outcomes provide weak supervision, this update should be interpreted as outcome-conditioned reliability rather than causal credit assignment.

Lifecycle filtering and capacity control. Experiences below a minimum confidence are removed from the active set, while entries unused for a fixed interval are archived. If the active library exceeds capacity B, entries are ranked by

$$S_{\text{life}}(e_j) = \lambda_{\text{life}} c_j + (1 - \lambda_{\text{life}}) \frac{n_j^+}{n_j^+ + n_j^- + 1}. \tag{6}$$

The lowest-scoring entries are suppressed from active retrieval. Thus, CSD determines which candidate knowledge is admitted, whereas CMU determines whether stored knowledge remains eligible for use.

3.5 Stage D: Decomposition and Retrieval

For a test task $x^\star = (q^\star, I^\star, \mathcal{U})$, an LLM decomposes the query into M method-oriented subqueries $g_1, \ldots, g_M$. Each subquery retrieves experience candidates using

$$S_{\text{ret}}(g_i, e_j) = \lambda_{\text{ret}} \cos(h_{g_i}, h_{e_j}) + (1 - \lambda_{\text{ret}}) c_j, \tag{7}$$

where h_{g_i} and h_{e_j} are embedding representations. Non-active entries and experiences with substantially more failures than successes are removed before ranking. Pairwise diversity filtering prevents near-duplicate items from filling the context. When two experiences from the same domain contain conflicting prescriptions, πSKILL retains the item with stronger confidence and empirical use history. The output of Stage D is therefore a sparse set of retrieved skills and top-k experiences for each subtask, matching panel D of Figure 2.

3.6 Stage E: Task Adaptation, Execution, and Feedback

Retrieved experiences are not inserted verbatim. A rewriting model conditions on the current query and visual input:

$$\widetilde{E} = R_\phi(q^\star, I^\star, E_{\text{ret}}), \tag{8}$$

and produces compact procedural guidance. The rewriter may discard an irrelevant experience but is instructed to preserve the operational content of retained items. The rewritten experiences are combined with a compact dataset-level skill prior and the base system prompt. Panel E also permits skill adaptation; in the reported configuration, this optional synthesis call is disabled and the retrieved compact skill is used directly. The combined guidance is injected into the agent execution loop. After the rollout terminates, the benchmark outcome and the identities of retrieved experiences are returned to the knowledge-version manager, which applies the update in Equation 5. This execution-to-repository path is the outer feedback loop shown at the bottom of Figure 2.

Algorithm 1 πSKILL accumulation and inference

Require: Accumulation tasks $\mathcal{D}_{\text{acc}}$, memory $\mathcal{M}$, test task $x^\star$
for $x \in \mathcal{D}_{\text{acc}}$ **do**
  Collect rollouts $\{\tau_i, y_i\}_{i=1}^{N}$
  $\{\bar{\tau}_i\}, S_x \leftarrow$ SUMMARIZEANDBUILDSKILL$(\{\tau_i, y_i\})$ ▷ A
  $E_{\text{core}} \leftarrow$ CROSSROLLOUTCRITIQUE$(\{\bar{\tau}_i, y_i\})$ ▷ B
  $E_{\text{local}} \leftarrow \bigcup_i C_\theta(x, \bar{\tau}_i, y_i)$ ▷ C
  Gate, deduplicate, and cap E_{local}
  Update skill library, experience bank, and version manager
end for
$G \leftarrow$ DECOMPOSE$(q^\star, I^\star)$ ▷ D
$(S_{\text{ret}}, E_{\text{ret}}) \leftarrow$ RETRIEVEANDDIVERSIFY$(G, \mathcal{M})$
$\widetilde{E} \leftarrow R_\phi(q^\star, I^\star, E_{\text{ret}})$ ▷ E
Run the agent with S_{ret}, $\widetilde{E}$, and the base prompt
Update confidence and lifecycle metadata from the execution outcome

4 EXPERIMENTS

Our experiments address four questions: **(RQ1)** Does πSkill improve multimodal-agent reliability across heterogeneous tasks? **(RQ2)** Which components account for the improvement? **(RQ3)** Does the method generalize across backbones and task distributions? **(RQ4)** How sensitive is the method to retrieval configuration and interaction budget?

4.1 BENCHMARKS AND EVALUATION PROTOCOL

Benchmarks. We evaluate on four benchmarks that emphasize different multimodal-agent capabilities. **VisualToolBench** Guo et al. (2025) tests visual reasoning with external tools, including image manipulation, code execution, and domain-specific inspection. **TIR-Bench** Li et al. (2025a) evaluates thinking-with-images tasks that require agents to transform, measure, or otherwise operate on visual evidence. **MMSearch-Plus** Tao et al. (2025) focuses on provenance-aware multimodal search. **AgentVista** Su et al. (2026) contains realistic visual scenarios that require multi-step interaction. Together, these benchmarks cover visual tool use, image-grounded reasoning, multimodal information seeking, and long-horizon agent behavior.

4.2 IMPLEMENTATION DETAILS

Unless otherwise stated, all controlled comparisons use Qwen2.5-VL-7B-Instruct Bai et al. (2025) as the agent backbone and bge-small-en-v1.5 Xiao et al. (2024) for experience retrieval. Generation uses temperature 0.3, top-$p = 1.0$, and at most 20 agent turns. Every evaluation task receives four independent rollouts. The available tools and their call limits are identical across methods.The default retrieval configuration uses a cosine-similarity threshold of 0.35, top-$k = 1$, and a 1,000-word skill budget. TIR-Bench uses threshold 0.5, top-$k = 5$, and a 300-word skill budget. The confidence-aware retrieval score uses $\lambda_{\text{ret}} = 0.6$. At inference time, retrieved experiences are rewritten for the current query and visual input, then combined with a compact dataset-level skill and the base system prompt. The optional LLM-based skill-adaptation call is disabled in the reported configuration; the retrieved compact skill is used directly. Appendix A provides the complete accumulation, lifecycle, retrieval, and evaluation settings.

4.3 BASELINES

We compare with tool-only and memory-based systems. **No Tools** performs direct inference without external tools or memory. **w/ Tools** uses the same tool interfaces as πSkill but no external memory; together, these two settings separate the contribution of tool access from that of memory.The memory-based baselines are: **AWM** Wang et al. (2024b), which induces reusable workflows from prior interactions; **Dynamic Cheatsheet (DC)** Suzgun et al. (2026), which maintains an adaptive test-time memory; **Agent-KB** Tang et al. (2025), which organizes cross-domain agent knowledge; and

Method	VisualToolBench		TIR-Bench		MMSearch-Plus		AgentVista		Average	
	A@4	P@4	A@4	P@4	A@4	P@4	A@4	P@4	A@4	P@4
No Tools	5.86	14.41	24.37	48.00	4.76	9.52	11.01	27.52	11.50	24.86
w/ Tools	8.78	23.42	25.50	53.00	2.14	6.19	9.17	19.27	11.40	25.47
AWM	10.70	25.23	27.75	48.00	3.81	10.95	8.49	20.18	12.69	26.09
DC	10.47	28.38	28.62	55.00	**7.14**	**15.71**	11.47	28.44	<u>14.43</u>	<u>31.88</u>
Agent-KB	10.81	24.78	<u>30.25</u>	<u>57.50</u>	<u>5.71</u>	<u>14.28</u>	9.86	23.85	14.16	30.10
XSkill	9.11	21.03	29.13	53.50	4.62	11.00	7.80	15.60	12.67	25.28
πSkill	**12.16**	**29.73**	**31.00**	**60.00**	<u>5.71</u>	13.33	**12.39**	**30.27**	**15.32**	**33.33**

Table 1: Main results with Qwen2.5-VL-7B-Instruct. A@4 and P@4 denote Average@4 and Pass@4. Bold and underlined values indicate the best and second-best results in each column.

XSkill Jiang et al. (2026), the closest baseline, which combines task-level skills with trajectory-derived experiences in multimodal agents. All memory-based methods receive the same 100 accumulation tasks and are evaluated under the protocol in Section 4.1.

4.4 Main Results

Aggregate performance. Table 1 shows that πSKILL obtains the highest macro-average among the evaluated methods. Compared with XSkill, it raises Average@4 from 12.67 to 15.32 and Pass@4 from 25.28 to 33.33. Compared with DC, the strongest alternative on both macro-average metrics, the gains are 0.89 and 1.45 percentage points.

Benchmark behavior. The improvement is heterogeneous. The largest gains over baseline occur on AgentVista, where Average@4 and Pass@4 increase by 4.59 and 14.67 points, and on VisualToolBench, where they increase by 3.05 and 8.70 points. Both benchmarks reward reusable procedural decisions across multi-step interactions. On MMSearch-Plus, πSkill improves over baseline but remains below DC. This result bounds the claim: the current memory pipeline is more effective for reusable operational procedures than for search tasks dominated by instance-specific evidence selection.

4.5 Component Analysis

We isolate three stages of the system: step-level knowledge extraction, lifecycle-aware memory maintenance, and inference-time contextualization. **Baseline+CSD** adds quality-gated single-rollout

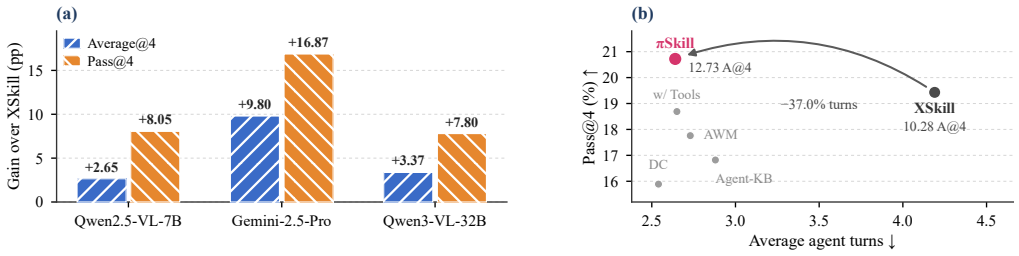


Figure 3: **Generalization and efficiency.** (a) Cross-backbone gains of πSkill over XSkill in Average@4 and Pass@4. (b) Separately recorded efficiency-controlled VisualToolBench results with Qwen3-VL-32B, showing Pass@4 against average agent turns.

step analysis to baseline. **Baseline+CMU** retains baseline's original knowledge construction but adds confidence and lifecycle maintenance. **πSkill w/o Decomp./Rewrite** removes task decomposition and experience rewriting from the full system. **πSkill w/o Memory** retains the same base agent and tool configuration but removes external experience memory.

Method	VisualToolBench		TIR-Bench		MMSearch-Plus		AgentVista		Average	
	A@4	P@4	A@4	P@4	A@4	P@4	A@4	P@4	A@4	P@4
Baseline	9.11	21.03	29.13	53.50	4.62	11.00	7.80	15.60	12.67	25.28
Baseline+CSD	9.00	23.36	**32.50**	**62.00**	4.62	12.50	6.88	11.93	13.25	27.45
Baseline+CMU	10.75	24.77	27.75	55.00	4.50	11.00	8.26	22.02	12.82	28.20
πSkill w/o Decomp./Rewrite	9.46	23.36	29.63	54.50	4.00	11.00	7.57	18.35	12.67	26.80
πSkill w/o Memory	10.70	29.28	30.12	59.00	5.60	**13.81**	10.32	22.02	14.19	31.03
πSkill	**12.16**	**29.73**	31.00	60.00	**5.71**	13.33	**12.39**	**30.27**	**15.32**	**33.33**

Table 2: Component analysis with Qwen2.5-VL-7B-Instruct. CSD and CMU are added independently to baseline; the final two variants test contextualized retrieval and external memory in the complete pipeline.

Extraction and maintenance are task dependent. CSD produces the strongest TIR-Bench result but degrades AgentVista, indicating that additional local experiences do not help every environment. CMU yields a smaller aggregate Average@4 change but improves AgentVista Pass@4 from 15.60 to 22.02. The complete system performs best on the macro-average, but the single-module results do not support a claim that either component is uniformly beneficial.

Contextualization is necessary. Removing decomposition and rewriting lowers aggregate Average@4 to 12.67, exactly the baseline level, and lowers Pass@4 from 33.33 to 26.80. The no-memory variant remains competitive, but the complete system improves it by 1.13 Average@4 and 2.30 Pass@4 points. The clearest conclusion from the ablation is therefore that stored experience must be selected and adapted to the current task before it becomes reliable guidance.

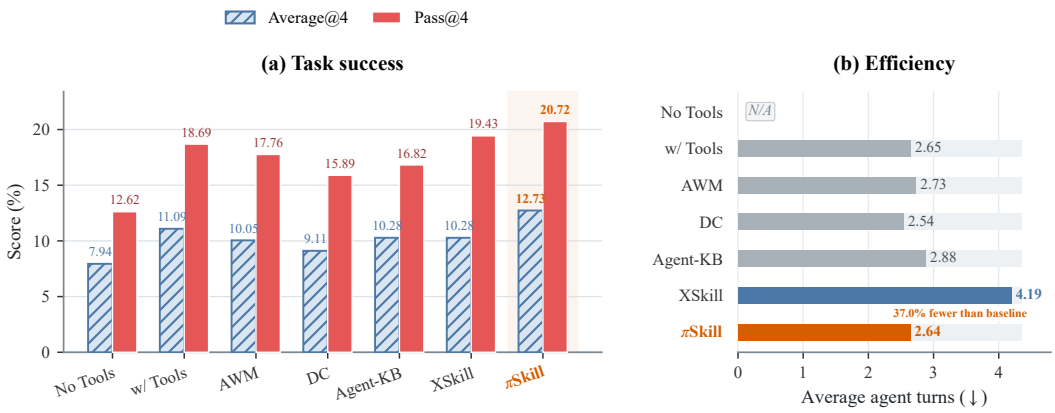


Figure 4: **Task success and interaction efficiency on VisualToolBench.** (a) Average@4 and Pass@4 across the evaluated tool and memory configurations. (b) Average agent turns for configurations with recorded interaction counts; lower is better. πSkill improves both success metrics while using 37.0% fewer turns than baseline. Exact values are reported in Appendix D.

4.6 Backbone Robustness and Cross-Task Transfer

Backbone robustness. Figure 3(a) summarizes the cross-backbone gains. Across Qwen2.5-VL-7B-Instruct, Gemini-2.5-Pro, and Qwen3-VL-32B, πSkill consistently improves the macro-average Average@4 and Pass@4 over baseline. The gain is largest with Gemini-2.5-Pro, increasing Average@4/Pass@4 from 17.93/31.15 to 27.73/48.02. With Qwen3-VL-32B, πSkill improves the same metrics from 10.47/18.86 to 13.84/26.66. The per-benchmark Qwen3-VL-32B results remain heterogeneous: the largest Pass@4 gain occurs on TIR-Bench, while AgentVista shows no Average@4 gain and a small Pass@4 decrease. Appendix C reports the complete numerical comparison.

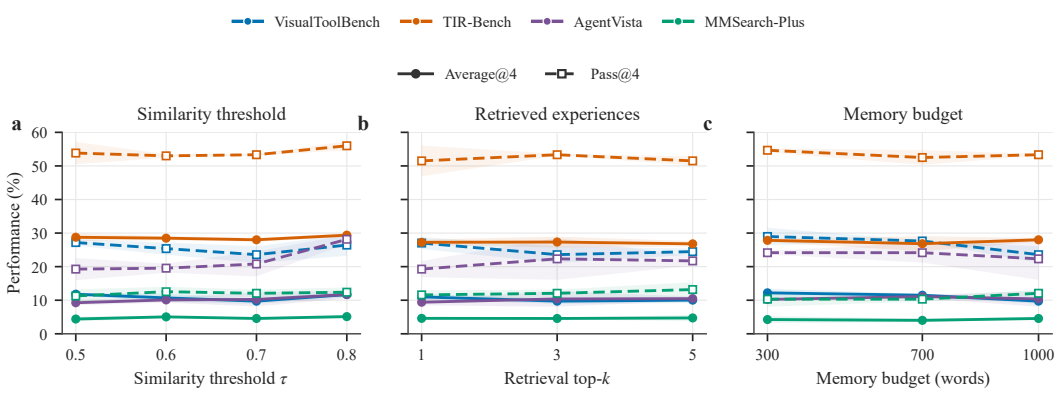


Figure 5: **Hyperparameter sensitivity of πSkill across four benchmarks.** Panels vary (a) the similarity threshold, (b) retrieval top-k, and (c) the skill budget. Solid lines with filled circles denote Average@4; dashed lines with open squares denote Pass@4. Shaded regions show mean $\pm$ one standard deviation over three runs.

Cross-task transfer. We construct memory on MMSearch-Plus and evaluate it on MMBrowseComp without target-benchmark accumulation. πSkill improves Baseline from 5.00/13.08 to 5.38/16.92 in Average@4/Pass@4, but remains below DC at 6.35/19.23. The result supports limited transfer beyond the source benchmark without showing universal cross-task generalization. Appendix B provides all method-level values.

4.7 Interaction Efficiency and Hyperparameter Sensitivity

We separately record interaction counts on VisualToolBench with Qwen3-VL-32B. Figure 3(b) provides the compact success–cost comparison, while Figure 4 reports all evaluated configurations. πSkill reaches 12.73 Average@4 and 20.72 Pass@4 with 2.64 average agent turns, compared with baseline's 10.28, 19.43, and 4.19 turns. This is a 37.0% reduction in agent interaction turns together with higher task success. The measurement excludes auxiliary decomposition and rewriting calls and should therefore be interpreted as *interaction efficiency*, not total token, latency, or compute efficiency. Appendix D reports the complete table.

We vary the semantic similarity threshold, retrieval top-k, and skill budget one at a time. Each configuration in the sensitivity study is averaged over three runs. Average@4 remains within a relatively narrow range, while Pass@4 is more variable. Within the tested range, threshold 0.8 gives the best macro-average in the similarity sweep. Increasing top-k from 1 to 5 provides no consistent benefit. A 300-word skill budget performs better than the 700- and 1,000-word alternatives on the macro-average. These observations suggest that relevance and compactness matter more than injecting a larger volume of memory. Complete per-benchmark values appear in Appendix E.

5 Conclusion

We presented πSkill, a training-free framework for extracting, maintaining, and contextualizing multimodal-agent experience. Following Figure 2, πSkill builds a lifecycle-aware repository from rollout-level and step-level knowledge, then retrieves and adapts relevant memories for new tasks. Across four benchmarks with Qwen2.5-VL-7B-Instruct, πSkill improves macro-average Average@4 from 12.67 to 15.32 and Pass@4 from 25.28 to 33.33 over the baseline. The gains are strongest on procedural multi-step tasks, while the ablations and transfer results highlight the importance of contextualized retrieval and the limits of universal experience reuse. Future work will improve credit assignment, long-term memory adaptation, and end-to-end efficiency.

AI USE STATEMENT

Generative AI tools were used for language editing and code debugging. They were not used to fabricate experimental data, select favorable results, verify citations, or make final scientific decisions. The authors manually reviewed AI-assisted text for technical accuracy and consistency with the reported evidence, inspected and tested AI-assisted code, and independently verified the results and references. The authors take responsibility for the complete content of the submission.

ETHICS STATEMENT

πSKILL stores and reuses model-generated operational experiences. Errors, biased heuristics, or unsafe action patterns in earlier trajectories may therefore propagate to later tasks. The memory may also retain task-specific information from tool interactions. Deployments should apply data minimization, access control, least-privilege tool permissions, and human oversight whenever trajectories may contain sensitive or proprietary information. Quality gating and lifecycle filtering reduce but do not eliminate these risks. πSKILL should not be used in medical, legal, financial, or other high-stakes settings without independent verification and task-specific safeguards.

REPRODUCIBILITY STATEMENT

Section 3 defines the extraction, maintenance, retrieval, and rewriting procedures in the same A–E order as Figure 2. Section 4 specifies the data split, backbone, rollout budget, metrics, and benchmark-specific retrieval settings. Appendix A lists all memory-update hyperparameters, while Appendices C–E provide extended numerical results. The anonymous supplement contains the core implementation, released benchmark splits, memory files, evaluation scripts, and configuration examples.

REFERENCES

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

Hanxiang Chao, Yihan Bai, Rui Sheng, Tianle Li, and Yushi Sun. Stale: Can llm agents know when their memories are no longer valid?, 2026. URL https://arxiv.org/abs/2605.06527.

Zheng Chu, Xiao Wang, Jack Hong, Huiming Fan, Yuqi Huang, Yue Yang, Guohai Xu, Chenxiao Zhao, Cheng Xiang, Shengchao Hu, et al. Redsearcher: A scalable and cost-efficient framework for long-horizon search agents. *arXiv preprint arXiv:2602.14234*, 2026.

Xinyu Geng, Peng Xia, Zhen Zhang, Xinyu Wang, Qiuchen Wang, Ruixue Ding, Chenxi Wang, Jialong Wu, Yida Zhao, Kuan Li, et al. Webwatcher: Breaking new frontier of vision-language deep research agent. *arXiv preprint arXiv:2508.05748*, 2025.

Xingang Guo, Utkarsh Tyagi, Advait Gosai, Paula Vergara, Jayeon Park, Ernesto Gabriel Hernández Montoya, Chen Bo Calvin Zhang, Bin Hu, Yunzhong He, Bing Liu, et al. Beyond seeing: Evaluating multimodal llms on tool-enabled image perception, transformation, and reasoning. *arXiv preprint arXiv:2510.12712*, 2025.

Zexue He, Yu Wang, Churan Zhi, Yuanzhe Hu, Tzu-Ping Chen, Lang Yin, Ze Chen, Tong Arthur Wu, Siru Ouyang, Zihan Wang, et al. Memoryarena: Benchmarking agent memory in interdependent multi-session agentic tasks. *arXiv preprint arXiv:2602.16313*, 2026.

Jack Hong, Chenxiao Zhao, ChengLin Zhu, Weiheng Lu, Guohai Xu, and Xing Yu. Deepeyesv2: Toward agentic multimodal model. *arXiv preprint arXiv:2511.05271*, 2025.

Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, et al. Memory in the age of ai agents. *arXiv preprint arXiv:2512.13564*, 2025.

Weiquan Huang, Zixuan Wang, Hehai Lin, Sudong Wang, Bo Xu, Qian Li, Beier Zhu, Linyi Yang, and Chengwei Qin. Ama: Adaptive memory via multi-agent collaboration. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 3099–3120, 2026.

Guanyu Jiang, Zhaochen Su, Xiaoye Qu, and Yi R Fung. Xskill: Continual learning from experience and skills in multimodal agents. *arXiv preprint arXiv:2603.12056*, 2026.

Chingkwun Lam, Jiaxin Li, Lingfei Zhang, and Kuo Zhao. Governing evolving memory in llm agents: Risks, mechanisms, and the stability and safety governed memory (ssgm) framework. *arXiv preprint arXiv:2603.11768*, 2026.

Ming Li, Jike Zhong, Shitian Zhao, Haoquan Zhang, Shaoheng Lin, Yuxiang Lai, Chen Wei, Konstantinos Psounis, and Kaipeng Zhang. Tir-bench: A comprehensive benchmark for agentic thinking-with-images reasoning. *arXiv preprint arXiv:2511.01833*, 2025a.

Shilong Li, Xingyuan Bu, Wenjie Wang, Jiaheng Liu, Jun Dong, Haoyang He, Hao Lu, Haozhe Zhang, Chenchen Jing, Zhen Li, et al. Mm-browsecomp: A comprehensive benchmark for multimodal browsing agents. *arXiv preprint arXiv:2508.13186*, 2025b.

Sijia Li, Yuchen Huang, Zifan Liu, Zijian Li, Lei Song, Jiang Bian, Jun Zhang, Rui Wang, et al. Experience-evolving multi-turn tool-use agent with hybrid episodic-procedural memory. *arXiv preprint arXiv:2512.07287*, 2025c.

Genglin Liu, Shijie Geng, Sha Li, Hejie Cui, Sarah Zhang, Xin Liu, and Tianyi Liu. Webcoach: Self-evolving web agents with cross-session memory guidance. *arXiv preprint arXiv:2511.12997*, 2025.

Jinghao Luo, Yuchen Tian, Chuxue Cao, Ziyang Luo, Hongzhan Lin, Kaixin Li, Chuyi Kong, Ruichao Yang, and Jing Ma. From storage to experience: A survey on the evolution of llm agent memory mechanisms. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 41622–41652, 2026a.

Z. C. Luo et al. Adarepair-mem: Adaptive experience orchestration for repository-level program repair, 2026b. URL `https://arxiv.org/abs/2609.20130`.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, et al. Reasoningbank: Scaling agent self-evolving with reasoning memory. *arXiv preprint arXiv:2509.25140*, 2025.

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.

Zhaochen Su, Jincheng Gao, Hangyu Guo, Zhenhua Liu, Lueyang Zhang, Xinyu Geng, Shijue Huang, Peng Xia, Guanyu Jiang, Cheng Wang, et al. Agentvista: Evaluating multimodal agents in ultra-challenging realistic visual scenarios. *arXiv preprint arXiv:2602.23166*, 2026.

Mirac Suzgun, Mert Yuksekgonul, Federico Bianchi, Dan Jurafsky, and James Zou. Dynamic cheatsheet: Test-time learning with adaptive memory. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7080–7106, 2026.

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.

Xijia Tao, Yihua Teng, Xinxing Su, Xinyu Fu, Jihao Wu, Chaofan Tao, Ziru Liu, Haoli Bai, Rui Liu, and Lingpeng Kong. Mmsearch-plus: Benchmarking provenance-aware search for multimodal browsing agents. *arXiv preprint arXiv:2508.21475*, 2025.

Chenyu Wang, Weixin Luo, Sixun Dong, Xiaohua Xuan, Zhengxin Li, Lin Ma, and Shenghua Gao. Mllm-tool: A multimodal large language model for tool agent learning. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 6678–6687. IEEE, 2025.

Jize Wang, Zerun Ma, Yining Li, Songyang Zhang, Cailian Chen, Kai Chen, and Xinyi Le. Gta: a benchmark for general tool agents. *Advances in Neural Information Processing Systems*, 37: 75749–75790, 2024a.

Shijian Wang, Jiarui Jin, Runhao Fu, Zexuan Yan, Xingjian Wang, Mengkang Hu, Eric Wang, Xiaoxi Li, Kangning Zhang, Li Yao, et al. Museagent: A multimodal reasoning agent with stateful experiences. *arXiv preprint arXiv:2603.27813*, 2026a.

Tong Wang, Pei Xu, Shiyue Cao, Likun Yang, Daipeng Li, Jianbin Jiao, and Kaiqi Huang. Samem: State-aware memory as a fine-grained memory for llm agents in decision-making. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 14691–14710, 2026b.

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

Rong Wu, Xiaoman Wang, Jianbiao Mei, Pinlong Cai, Daocheng Fu, et al. Evolver: Self-evolving llm agents through an experience-driven lifecycle. *arXiv preprint arXiv:2510.16079*, 2025.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings, 2024. URL `https://arxiv.org/abs/2309.07597`.

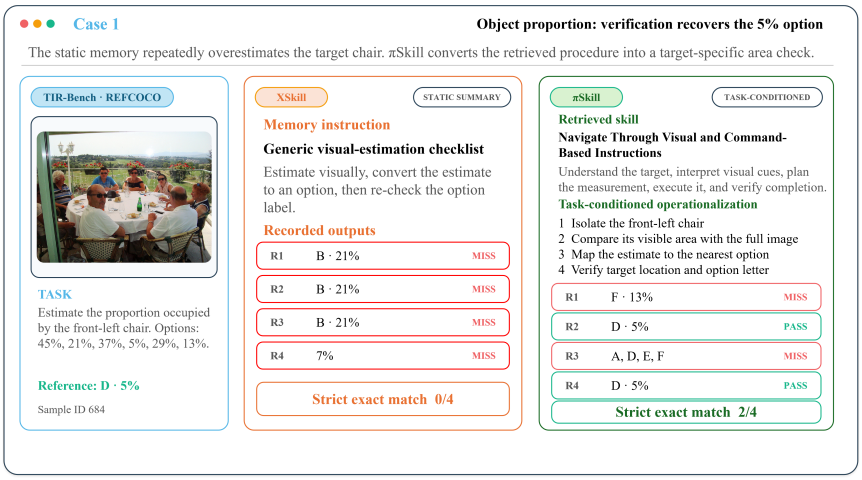
Zidi Xiong, Yuping Lin, Wenya Xie, Pengfei He, Zirui Liu, Jiliang Tang, Himabindu Lakkaraju, and Zhen Xiang. How memory management impacts llm agents: An empirical study of experience-following behavior. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 623–645, 2026.

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.

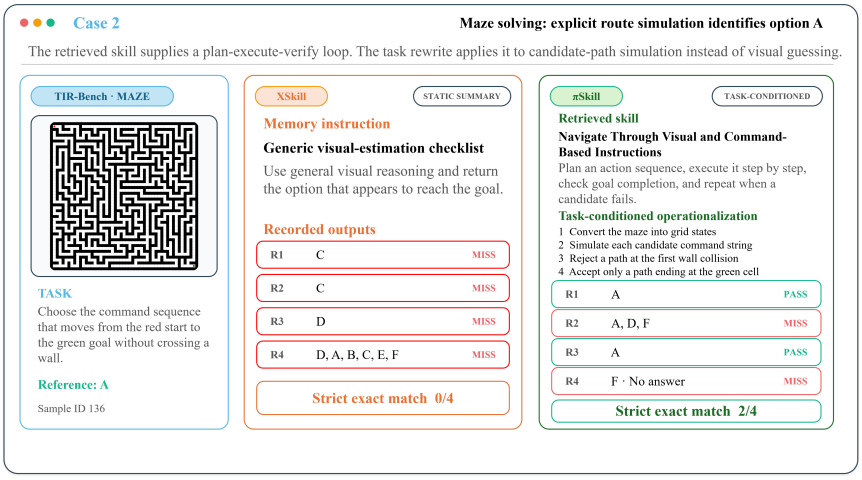
Neeraj Yadav. Temporal validity in retrieval memory: Eliminating stale-fact errors for ai agents over evolving knowledge. *arXiv preprint arXiv:2606.26511*, 2026.

Xing Zhang, Guanghui Wang, Yanwei Cui, Wei Qiu, Ziyuan Li, Bing Zhu, and Peiyang He. Experience compression spectrum: Unifying memory, skills, and rules in llm agents. *arXiv preprint arXiv:2604.15877*, 2026.

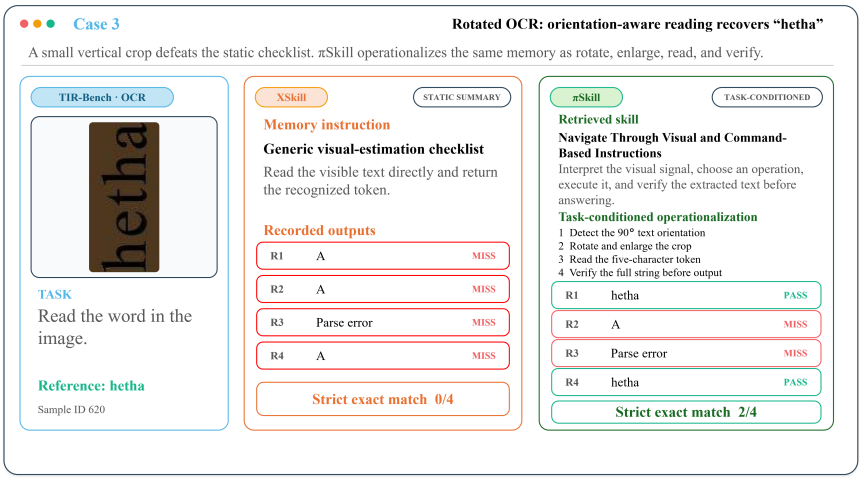
Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.



(a) Object proportion estimation (Sample ID 684).



(b) Maze navigation (Sample ID 136).



(c) Rotated text recognition (Sample ID 620).

Figure 6: Qualitative case studies on TIR-Bench comparing baseline and πSkill across three multimodal reasoning tasks. (a) Object proportion estimation. (b) Maze navigation. (c) Rotated text recognition. In all three cases, πSkill achieves 2/4 strict exact matches, compared with 0/4 for baseline.

A Implementation and Reproducibility Details

A.1 Evaluation Configuration

Backbones and decoding. Unless otherwise stated, the primary experiments use Qwen2.5-VL-7B-Instruct as the multimodal agent backbone. Open-source models are served with vLLM, and the primary experiments are executed on four NVIDIA RTX PRO 6000 GPUs. Generation uses temperature 0.3, top-$p = 1.0$, and a maximum of 20 agent turns. Each evaluation task receives four independent rollouts. Gemini-2.5-Pro and Qwen3-VL-32B are evaluated with the same benchmark splits, tool interfaces, and rollout protocol used in the primary comparison.

Data separation. For each benchmark, 100 tasks are reserved exclusively for memory accumulation and all remaining tasks are used for evaluation. The accumulation and evaluation sets are disjoint. All methods use the same task split, task order, available tools, and per-task interaction limit. Memories constructed on the accumulation partition remain fixed with respect to evaluation answers: only the completed-task success signal is available to the lifecycle update after an evaluation rollout terminates.

Metrics. For task i and rollout r, let $y_{i,r} \in \{0, 1\}$ denote success. Average@4 is the mean of $\frac{1}{4}\sum_{r=1}^{4} y_{i,r}$ over evaluation tasks, whereas Pass@4 is the mean of $\mathbb{I}[\max_r y_{i,r} = 1]$. Reported "Average" columns are unweighted macro-averages across the four benchmarks. All comparisons in the paper use absolute percentage-point differences.

A.2 Knowledge Construction and Retrieval

Embedding and retrieval. Experience retrieval uses bge-small-en-v1.5 and cosine similarity. The default configuration uses a similarity threshold of 0.35, top-$k = 1$, and a 1,000-word skill budget. TIR-Bench uses threshold 0.5, top-$k = 5$, and a 300-word skill budget. The confidence-aware retrieval score uses $\lambda_{\text{ret}} = 0.6$, and the maximum permitted pairwise similarity among selected experiences is 0.82. The optional LLM-based skill-adaptation call is disabled in the reported configuration; the compact retrieved skill is used directly, while retrieved experiences are rewritten for the current task.

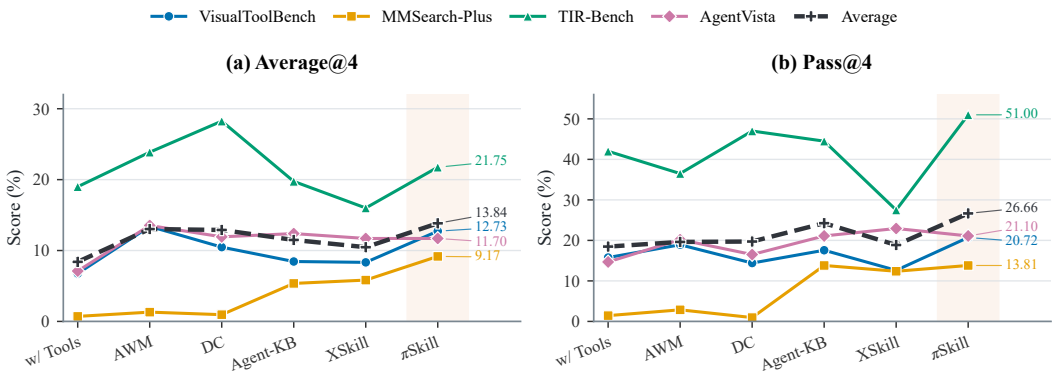


Figure 7: **Per-benchmark results with Qwen3-VL-32B.** Average@4 and Pass@4 are shown for the six retained tool-enabled and memory-based methods. Appendix C reports the exact values.

Local-experience admission. The quality score in Equation 3 starts at $q_0 = 0.50$. It adds 0.15 for candidates containing 12–80 words, 0.15 for an explicit trigger condition, and 0.15 for an executable operation. It subtracts 0.25 for fewer than 8 words, 0.10 for more than 120 words, 0.30 for sample-specific content, and 0.20 for vague advice. Scores are clipped to $[0, 1]$, and candidates are admitted when $q(e) \geq \eta = 0.72$. At most one local experience is retained per rollout group. Within each update batch, local experiences are capped at 30% of the number of core experiences and at 32 items in absolute terms.

Sweep	Value	Sim.	Top-k	Words	VisualToolBench	TIR-Bench	AgentVista	MMSearch-Plus	Average
Similarity	0.5	0.5	3	1000	11.79/27.18	28.75/53.83	9.25/19.27	4.40/11.27	13.55/27.89
Similarity	0.6	0.6	3	1000	10.70/25.38	28.48/53.00	10.09/19.54	5.04/12.54	13.58/27.61
Similarity	0.7	0.7	3	1000	9.72/23.57	28.00/53.33	10.24/20.79	4.56/12.06	13.13/27.44
Similarity	0.8	0.8	3	1000	11.67/26.43	29.38/56.00	11.70/28.14	5.12/12.38	**14.47/30.74**
Top-k	1	0.7	1	1000	11.00/27.03	27.25/51.50	9.40/19.27	4.60/11.59	13.06/27.35
Top-k	3	0.7	3	1000	9.72/23.57	27.33/53.33	10.40/22.32	4.56/12.06	13.00/27.82
Top-k	5	0.7	5	1000	10.02/24.47	26.79/51.50	10.47/21.71	4.72/13.17	13.00/27.71
Budget	300	0.7	3	300	12.20/28.98	27.83/54.67	10.24/24.16	4.25/10.32	**13.63/29.53**
Budget	700	0.7	3	700	11.49/27.63	26.83/52.50	11.09/24.16	4.01/10.32	13.36/28.65
Budget	1000	0.7	3	1000	9.72/23.57	28.00/53.33	10.40/22.32	4.56/12.06	13.17/27.82

Table 3: Three-run mean sensitivity results. Each benchmark entry and the final average report Average@4/Pass@4. The repeated $(0.7, 3, 1000)$ configuration is measured independently within each sweep.

Lifecycle parameters. New experiences are initialized with confidence $c_{\text{init}} = 0.7$. A successful attributed use increases confidence by $\Delta_+ = 0.05$, whereas a failed use decreases it by $\Delta_- = 0.10$. An experience is marked `to_review` when it has at least three failed uses and more failures than successes. Entries with confidence below 0.3 are removed from the active set, and entries unused for 30 days are archived. When capacity control is required, the lifecycle score uses $\lambda_{\text{life}} = 0.6$.

A.3 Reporting and Traceability

The released experiment record should contain the exact benchmark split files, model identifiers, decoding settings, tool configuration, retrieval parameters, memory snapshot, and evaluation outputs used for every reported row. For each evaluation rollout, the record should also preserve the retrieved experience identifiers, rewritten guidance, terminal answer, success label, and number of agent turns. This information is sufficient to recompute Average@4, Pass@4, and interaction-turn statistics and to audit which memory items were available to the agent. Together, these records make the complete memory lifecycle transparent and reproducible, from experience retrieval and task-conditioned rewriting to confidence updating and final execution.

B Cross-Task Transfer

We construct memory on MMSearch-Plus and evaluate on MMBrowseComp without rebuilding memory from target-benchmark tasks. This setting tests whether the stored procedures transfer beyond the distribution on which they were accumulated.

Method	Average@4	Pass@4
AWM	4.42	12.31
DC	**6.35**	**19.23**
Agent-KB	4.81	16.15
XSkill	5.00	13.08
XSkill+CSD	4.62	13.08
XSkill+CMU	4.23	13.84
πSkill	<u>5.38</u>	<u>16.92</u>

Table 4: Cross-task transfer from MMSearch-Plus memory to MMBrowseComp with Qwen2.5-VL-7B-Instruct. Bold and underlined values indicate the best and second-best results.

Without using any target-benchmark accumulation tasks, πSkill reaches 5.38 Average@4 and 16.92 Pass@4, ranking second among all evaluated methods on both metrics. Its 3.84-point Pass@4 improvement over the trajectory-level memory implementation demonstrates that the learned procedures remain useful after a direct distribution shift. The complete system also outperforms either independently augmented variant, showing that fine-grained extraction, lifecycle-aware maintenance, and task-conditioned reuse are most effective when deployed as a coordinated pipeline.

C Additional Backbone Results

Backbone	Method	Average@4	Pass@4	Avg. turns
Gemini-2.5-Pro	No Tools	16.33	27.35	1.00
	w/ Tools	22.23	40.45	3.84
	AWM	13.18	22.89	1.00
	DC	17.14	29.30	1.00
	Agent-KB	23.82	40.96	4.38
	XSkill	17.93	31.15	4.30
	πSkill	**27.73**	**48.02**	5.06
Qwen3-VL-32B	No Tools	14.26	28.11	1.00
	w/ Tools	8.40	18.47	1.91
	AWM	13.03	19.62	1.00
	DC	12.90	19.72	1.00
	Agent-KB	11.49	24.25	2.37
	XSkill	10.47	18.86	3.91
	πSkill	**13.84**	**26.66**	3.02
Qwen3-VL-7B	πSkill	12.88	21.62	3.07

Table 5: Macro-average results with additional backbones. Turns are averaged across the four benchmarks.

The additional-backbone results demonstrate that πSkill transfers effectively across model scales and families. It achieves the strongest macro-average Average@4 and Pass@4 with both Gemini-2.5-Pro and Qwen3-VL-32B, reaching 27.73/48.02 and 13.84/26.66, respectively. At the benchmark level, πSkill obtains the best Pass@4 on VisualToolBench and TIR-Bench and the best Average@4 on MMSearch-Plus with Qwen3-VL-32B. These results show that the framework provides a model-agnostic mechanism for converting accumulated multimodal experience into stronger downstream behavior.

D Interaction-Efficiency Results

Method	Average@4	Pass@4	Avg. turns
No Tools	7.94	12.62	–
w/ Tools	11.09	18.69	2.65
AWM	10.05	17.76	2.73
DC	9.11	15.89	2.54
Agent-KB	10.28	16.82	2.88
XSkill	10.28	19.43	4.19
πSkill	**12.73**	**20.72**	2.64

Table 6: Efficiency-controlled VisualToolBench results with Qwen3-VL-32B. Average turns count agent interactions and exclude auxiliary decomposition and rewriting calls.

πSkill achieves the strongest Average@4 and Pass@4 in the efficiency-controlled evaluation while requiring only 2.64 agent turns. It therefore combines a 37.0% reduction in interaction turns relative to the trajectory-level memory implementation with gains of 2.45 Average@4 and 1.29 Pass@4 points. This result shows that retrieved and contextualized experience helps concentrate the interaction budget on productive decisions rather than relying on longer trajectories.

E Hyperparameter Sensitivity

Across all tested configurations, πSkill maintains a stable macro-average, demonstrating robustness to the similarity threshold, retrieval top-k, and skill budget. A threshold of 0.8 reaches the strongest result in the similarity sweep at 14.47 Average@4 and 30.74 Pass@4. Strong performance is also preserved with only one retrieved experience, while the compact 300-word setting achieves

Method	VisualToolBench		MMSearch-Plus		TIR-Bench		AgentVista		Average	
	A@4	P@4	A@4	P@4	A@4	P@4	A@4	P@4	A@4	P@4
w/ Tools	6.76	15.77	0.71	1.43	19.00	42.00	7.11	14.68	8.40	18.47
AWM	**13.40**	18.92	1.31	2.86	23.87	36.50	**13.53**	20.18	13.03	19.62
DC	10.48	14.41	0.95	0.95	**28.25**	47.00	11.93	16.51	12.90	19.72
Agent-KB	8.45	17.57	5.36	**13.81**	19.75	44.50	12.39	21.10	11.49	24.25
XSkill	8.33	12.61	5.83	12.38	16.00	27.50	11.70	**22.94**	10.47	18.86
πSkill	12.73	**20.72**	**9.17**	**13.81**	21.75	**51.00**	11.70	21.10	**13.84**	**26.66**

Table 7: Complete per-benchmark results with Qwen3-VL-32B. The same values are visualized in Figure 7.

the best budget-sweep average at 13.63 Average@4 and 29.53 Pass@4. These results highlight a practical advantage of πSkill: selectively retrieved, compact knowledge is sufficient to deliver strong performance without requiring a large memory context.

F Qualitative Case Studies

Object-proportion estimation. For object-proportion estimation, πSkill converts a general visual-estimation procedure into a task-specific sequence of target localization, area comparison, option mapping, and verification. This operationalization recovers the correct 5% option in two of four rollouts, compared with 0/4 for the static-memory baseline.

Maze navigation. For maze navigation, πSkill instantiates the retrieved planning procedure as an explicit grid-state simulation: parse the maze, test candidate command strings, reject a path at the first collision, and accept only a route that reaches the goal. This task-conditioned plan–execute–verify loop raises strict exact-match performance from 0/4 to 2/4.

Rotated-text recognition. For rotated-text recognition, πSkill grounds the retrieved procedure in the image orientation and produces a concrete sequence of rotation detection, crop enlargement, OCR, and full-string verification. The resulting guidance recovers the reference token "hetha" in two of four rollouts, again improving over the 0/4 static-memory result.

Together, these cases make the general mechanism of πSkill directly observable: retrieval supplies reusable procedural knowledge, while task-conditioned rewriting transforms that knowledge into an executable strategy grounded in the current visual input. The consistent 0/4-to-2/4 improvement across three distinct task types provides qualitative evidence that contextualization is a key source of the framework's effectiveness.