πSkill: High-Density Knowledge Extraction from Single Trajectories via Circular Step-Level Analysis

Anonymous ACL submission

Abstract

Multimodal agents can solve complex reasoning tasks with external tools, but still struggle to improve from past trajectories without parameter updates. Existing training-free memory methods often rely on coarse trajectory-level summaries and static repositories, limiting fine-grained failure analysis and memory reliability. To address this, we propose **πSkill**, a lifecycle-managed continual learning framework for multimodal agents. Its Circular Step-level Knowledge Distillation (CSD) extracts high-density experience atoms from successful and failed step transitions, while Confidence-aware Lifecycle Memory Updating (CMU) tracks confidence, usage statistics, lifecycle states, and version history to refine or suppress unreliable knowledge. During inference, πSkill retrieves knowledge through semantic recall and confidence-aware refinement, with execution feedback used for online memory updating. Experiments with Qwen2.5-VL-7B-Instruct on four multimodal agent benchmarks show that πSkill improves the average Average@4 from 12.41 to 14.02 and Pass@4 from 26.53 to 29.73, demonstrating the effectiveness of fine-grained knowledge distillation and lifecycle-aware memory evolution.

1 Introduction

The evolution of Multimodal Large Language Models (MLLMs) has transformed autonomous agents from passive perceptual systems into active problem solvers. In open-ended settings, multimodal agents integrate visual perception, code execution, and web browsing to solve complex multi-step tasks. They are increasingly expected to orchestrate heterogeneous tools over long horizons (Guo et al., 2025; Chu et al., 2026). To handle changing environments, agents must continuously adapt their internal workflows (Geng et al., 2025). However, parameter fine-tuning or reinforcement learning is costly, sample-inefficient, and may disrupt the alignment of foundation models.

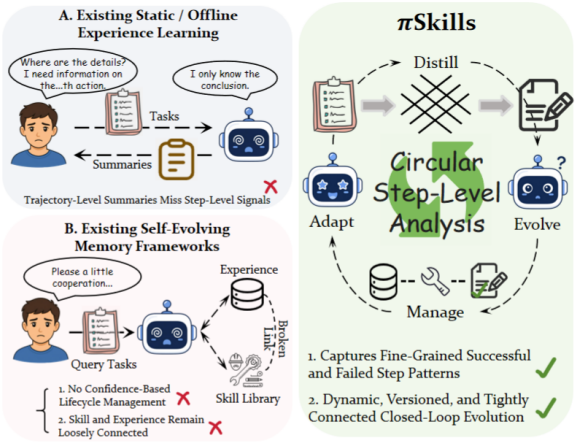

Figure 1: Comparison of experience-learning paradigms. Static/offline methods rely on trajectory-level summaries and thus miss fine-grained step-level signals. Existing self-evolving memory frameworks lack confidence-aware lifecycle management and maintain weak links between experiences and skills. In contrast, πSkill performs circular step-level analysis over successful and failed transitions, enabling dynamic, versioned, and tightly connected closed-loop evolution for multimodal agents.

Training-free, non-parametric continual learning from past trajectories has therefore become a promising paradigm (Wu et al., 2025). Similar to human problem solving, recent agents store both action-level experiences and task-level skills (Ouyang et al., 2025). For example, Agent Workflow Memory (AWM) maintains external repositories to guide future execution (Wang et al., 2024). However, as shown in Figure 1, existing experience-driven agents still rely heavily on holistic trajectory summaries and text-based logs, leading to three key bottlenecks:

- **Information Sparsity in Knowledge Generation:** Existing methods often summarize an entire rollout as a coarse success or failure (Suzgun et al., 2025), ignoring step-level dynamics within individual traces.

Thus, trajectory-level retrieval provides limited help for intermediate bottlenecks, where state-specific tactical knowledge is more useful (Wang et al., 2026; Xie et al., 2024). This reduces information density and requires large amounts of rollout data.

- **Rigidity in Repository Storage:** Extracted skills and experiences are usually stored as markdown files or flat JSON registries, without version control, metadata tracking, or relational management (Hu et al., 2025). They also lack effective cross-level compression and dynamic memory linking (Zhang et al., 2026). Without lifecycle management, repositories cannot roll back corrupted knowledge or organize memories into evolving networks (Liu et al., 2025), causing semantic drift and redundancy.

- **Vulnerability in Static Retrieval:** Current inference mainly retrieves memories through textual or semantic similarity, ignoring their reliability and confidence. As a result, stale or low-quality memories may amplify downstream errors (Lam et al., 2026). Static retrieval is also disconnected from real-time execution feedback, limiting online self-correction and active memory testing (He et al., 2026).

To address these bottlenecks, we introduce πSkill, a lifecycle-managed continual learning framework that explicitly centers on two core modules: Circular Step-level Knowledge Distillation (CSD) and Confidence-aware Lifecycle Memory Updating (CMU). As shown in Figure 2, πSkill shifts non-parametric agent learning from coarse trajectory-level summarization to fine-grained step-level knowledge extraction, and further turns static memory repositories into dynamically updated and reliability-aware knowledge bases. Under a standard multi-path rollout setting ($N = 4$), the framework consists of three tightly connected stages.

We evaluate πSkill with Qwen2.5-VL-7B-Instruct on four multimodal agent benchmarks: VisualToolBench (Guo et al., 2025), TIR-Bench (Li et al., 2025a), MMSearch-Plus (Tao et al., 2025), and AgentVista (Su et al., 2026). Experimental results show that πSkill consistently improves both Average@4 and Pass@4 over XSkill. Ablation studies further verify that CSD and CMU are complementary: CSD improves the granularity of experience generation, while CMU enhances the reliability and adaptability of the memory bank.

Our contributions are summarized as follows:

- **Granular Knowledge Generation:** We propose step-level distillation to extract dense knowledge from local successful and failed transitions instead of coarse trajectory summaries.

- **Lifecycle-Managed Repositories:** We build a version-controlled and relationally linked storage architecture with confidence tracking and skill-experience linking to reduce repository degradation.

- **Two-Layer Online Closed Loop:** We design a two-layer retrieval mechanism that combines semantic recall with confidence refinement and feeds execution outcomes back for continual memory updating.

2 Related Works

Multimodal Tool-Augmented Agents. Recent multimodal agents have shifted from passive perception to active problem solving. By invoking external tools such as code execution, image processing, web search, image search, and webpage browsing, these agents can solve complex tasks that require multi-step visual reasoning and interaction with external environments (Li et al., 2025b). Recent benchmarks further evaluate these capabilities from tool-enabled image reasoning, thinking-with-images tasks, and realistic long-horizon multimodal scenarios (Guo et al., 2025; Li et al., 2025a; Su et al., 2026). However, existing multimodal agents still suffer from inefficient tool use and unstable tool orchestration. Although recent systems attempt to enhance multimodal tool reasoning and capability composition (Hong et al., 2025), most agents still solve each task as an isolated episode, causing useful intermediate decisions, tool-use patterns, and failure feedback to be discarded after execution. Inspired by recent studies on persistent multimodal memory (Liu et al., 2025), our work explores a training-free agent improvement approach that transforms historical trajectories into structured, reusable, and dynamically maintained knowledge.

Learning from Experience, Skills, and Continual Memory. Learning from historical trajectories has become an important direction for improving

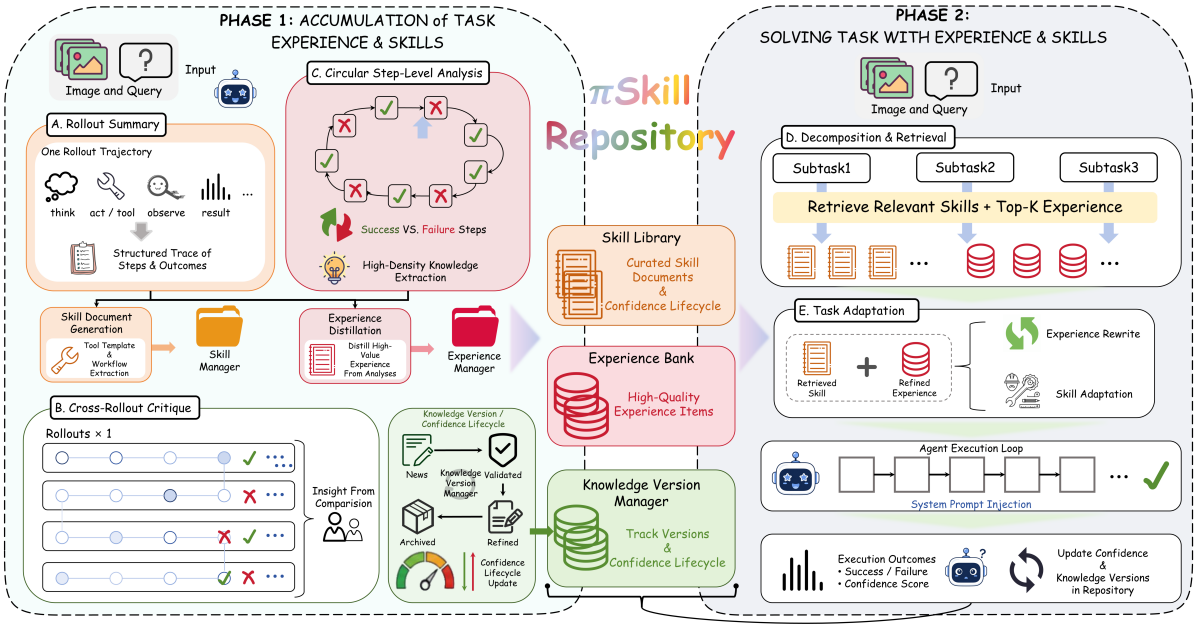

Figure 2: Overview of the πSkill framework. Phase 1 (left): Via (A) single-rollout summarization, (B) cross-rollout critique and (C) circular step-level analysis, the agent mines structured skills and high-value experiences from multi-trajectory interactions to update its knowledge repository dynamically. Phase 2 (right): For new tasks, it retrieves matched knowledge by (D) task decomposition retrieval, conducts adaptation via (E) skill-experience adjustment, and feeds optimized knowledge into the agent execution loop; execution feedback further updates repository knowledge versions and confidence metrics.

agent capabilities. Existing methods usually transform past interactions into experiences, workflows, skills, or reasoning memories, such as extracting lessons from previous trials (Zhao et al., 2024), inducing reusable workflows (Wang et al., 2024), maintaining adaptive test-time memory (Suzgun et al., 2026), or organizing cross-domain experience for planning and diagnosis (Tang et al., 2025). In multimodal settings, recent work also extracts task-level skills and action-level experiences from visual-tool interactions through visually grounded trajectory summarization and cross-rollout critique (Jiang et al., 2026). Although these methods demonstrate the effectiveness of experience and skill memories, most of them still rely on trajectory-level abstraction, which may overlook the internal heterogeneity of a single trajectory. Meanwhile, as memory scales up, stored knowledge may become redundant, outdated, or unreliable (Ouyang et al., 2025; Wu et al., 2025). In contrast, πSkill performs step-level knowledge distillation from a single trajectory, extracting positive experiences, failure-aware lessons, and reusable operational patterns through local contrastive analysis. It further maintains confidence, version history, and skill-experience relations through lifecycle-aware knowledge management, allowing useful knowledge to be strengthened while unreliable knowledge is revised, archived, or rolled back.

3 Method

The πSkill framework introduces a high-density knowledge extraction and management ecosystem designed to continuously distill, evolve, and utilize step-level agent experiences. As illustrated in the architecture diagram, **πSkill** breaks away from conventional trajectory-level heuristics by implementing a closed-loop system across three decoupled yet interconnected subsystems:

- Knowledge Generation (Training Phase): Performs fine-grained step-level knowledge distillation using both single-trajectory critiques and cross-trajectory contrastive analysis.
- Knowledge Storage and Lifecycle Management (Storage Layer): Orchestrates dynamic state updates, relational skill-experience mapping, and version-controlled rollbacks.
- Adaptive Retrieval and Inference (Inference Phase): Executes a dual-tier decomposition retrieval strategy coupled with experience rewriting, feeding runtime outcomes back into

the storage layer to complete the circular optimization loop.

3.1 Circular Step-level Knowledge Distillation (CSD)

Circular Step-level Knowledge Distillation (CSD) aims to extract dense and localized knowledge from agent trajectories. Unlike traditional methods that mainly summarize complete rollouts into broad heuristics, CSD decomposes each trajectory into step transitions and analyzes how local actions contribute to final success or failure. This design allows πSkill to obtain useful experience atoms even from a limited number of rollouts.

Multimodal Rollout and Summarization Let a multimodal task sample be denoted as $X = (Q, I, \mathcal{U})$, where Q represents the textual query, I denotes accompanying visual inputs, and $\mathcal{U}$ represents available external tools. During the training phase, the agent executes multiple rollouts, yielding a set of execution trajectories $\mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_N\}$. Each trajectory τ is represented as a sequential chain of steps:

$$\tau = \{s_1, a_1, s_2, a_2, \ldots, s_T\} \tag{1}$$

where s_t denotes the environment state at step t and a_t denotes the action or tool call executed. These raw trajectories are consolidated via a trajectory summarization module to form a foundational corpus for distillation.

Single-Trajectory Step Critique in CSD To mitigate the heavy dependency on massive, diverse rollout collections, πSkill introduces a localized critique mechanism (single_rollout_critique). For any standalone trajectory τ, each step transition (s_t, a_t, s_{t+1}) is analyzed in isolation against the ultimate global task outcome. The model isolates successful transitions from failure points to derive a step-level critique:

$$C_{\text{step}} = \text{Critique}(s_t, a_t, s_{t+1} \mid X, \text{Outcome}(\tau)) \tag{2}$$

This step-level distillation forces the model to extract high-density lessons even from a single, unvaried trajectory execution by identifying the exact sub-step responsible for success or failure.

Cross-Trajectory Contrastive Enhancement Simultaneously, when multiple rollouts are available, an intra-sample contrastive analysis module (intra_sample_experiences) contrasts successful trajectories τ^+ against failing trajectories τ^- originating from the exact same sample X. By performing cross-trajectory step alignment, the system pinpoints pivotal divergence nodes:

$$\Delta_{\text{step}} = \text{Align}(\tau^+, \tau^-) \tag{3}$$

This dual-pathway approach generates an Experience Atom $\mathcal{E}$, formally defined as a tuple:

$$\mathcal{E} = \langle c, \text{domain}, \text{lesson}, \text{tool\_template}, \phi_{\text{succ}}, \phi_{\text{fail}} \rangle \tag{4}$$

where $c \in [0, 1]$ is the scalar confidence score, domain defines the task scope, lesson represents the distilled natural language rule, tool_template abstracts the action format, and $\phi_{\text{succ}}, \phi_{\text{fail}}$ maintain historical execution counts. The finalized atoms are transformed into explicit executable code structures or prompt constraints.

3.2 Confidence-aware Lifecycle Memory Updating (CMU)

Confidence-aware Lifecycle Memory Updating (CMU) turns the extracted experience atoms into an evolving and reliability-aware memory bank. Instead of treating stored experiences as immutable prompts, CMU continuously tracks how each experience performs during later inference. Each experience is associated with confidence, success and failure counts, lifecycle status, and version metadata, allowing the system to reinforce useful knowledge while suppressing or archiving unreliable entries.

Experience-Skill Relational Mapping in CMU The storage layer splits structural knowledge into two core repositories: the Experience Database (experiences.json, storing contextual lessons $\mathcal{E}$) and the Skill Base (skills/, storing reusable tool wrappers and macro-actions $\mathcal{S}$). The SkillBuilder module acts as a relational bridge, binding experience atoms to atomic or compound skills via specific identifier mappings:

$$\mathcal{M} : \mathcal{E} \rightarrow \{\mathcal{S}_{\text{id1}}, \mathcal{S}_{\text{id2}}, \ldots\} \tag{5}$$

This ensures that whenever a skill is invoked, its accompanying contextual experiences, failure warnings, and operational boundaries are retrieved simultaneously.

Dynamic Confidence and State Updates Rather than treating knowledge as immutable, πSkill implements real-time state updates based on empirical execution records. Each experience atom possesses

a dynamic state vector $\mathcal{V} = \langle \phi_{\text{succ}}, \phi_{\text{fail}}, \text{status} \rangle$. The confidence score c is calculated and dynamically adjusted according to:

$$c = \alpha \cdot c_{\text{prior}} + (1 - \alpha) \cdot \left(\frac{\phi_{\text{succ}}}{\phi_{\text{succ}} + \phi_{\text{fail}} + \epsilon} \right) \quad (6)$$

where $\alpha \in [0, 1]$ is a temporal decay hyperparameter and ϵ is a smoothing factor. Based on this confidence value, the status of an experience transitions along a standardized lifecycle (e.g., Candidate, Active, or Archived).

Version Control and Rollback Mechanics To ensure safe memory consolidation, a dedicated KnowledgeVersionManager tracks changes across the metadata schemas and version folders. The version manager implements three distinct routines: **init**$(\cdot)$: Establishes a new knowledge base baseline. **archive**$(\cdot)$ / **new_version**$(\cdot)$: Commits a snapshot of the current state after a training epoch or an online deployment cycle. **rollback**$(\cdot)$: If newly injected experiences cause an optimization drop or introduce systemic hallucinations during execution, the system rolls back to a stable ancestral state $\mathcal{K}_{t-k}$.

3.3 Confidence-aware Retrieval and Circular Feedback

During the inference phase, πSkill leverages a nonlinear, adaptive retrieval pipeline that maps new tasks to historical experiences, executing actions within a circular feedback loop.

Dual-Tier Experience Retrieval Given a novel, complex incoming task $X_{\text{test}} = (Q_{\text{test}}, I_{\text{test}})$, relying purely on surface-level embedding similarities frequently results in suboptimal knowledge matching. πSkill overrides this via a two-tier ExperienceRetriever:

Task Decomposition Retrieval: The master task X_{test} is decomposed into a set of sequential subgoals $\{g_1, g_2, \ldots, g_m\}$. A coarse semantic recall is then executed over the experience base for each sub-goal individually using a bi-encoder similarity score:

$$\text{Sim}(g_i, \mathcal{E}) = \cos(\mathbf{e}_{g_i}, \mathbf{e}_{\mathcal{E}}) \quad (7)$$

Confidence Refinement and Conflict Resolution: The recalled experiences are filtered by their operational confidence c. If two retrieved experiences $\mathcal{E}_A$ and $\mathcal{E}_B$ provide conflicting guidelines for a given sub-goal, the system resolves the conflict by favoring the atom with the higher confidence score or the more specific structural domain match.

Experience Rewriting and Skill Adaptation Once the optimal experience atoms are isolated, they pass through an inline task-adaptive rewriting module. This module strips away training-specific artifacts from the raw lesson, reshaping the experience text to perfectly match the target constraints of X_{test}:

$$\mathcal{E}^* = \text{Rewrite}(\mathcal{E} \mid X_{\text{test}}) \quad (8)$$

Following experience rewriting, the SkillBuilder passes the relevant skill pointers to the adaptation engine, customizing the concrete execution templates to accommodate the current test inputs.

Agent Execution and the Circular Loop The adapted skills and rewritten experiences are injected directly into the Agent's context window. The agent executes the plan using tool calls (Agent Execution / Tool Use), generating an inference trajectory τ_{eval} that terminates in a verifiable outcome (Success or Failure).Crucially, this outcome is not discarded. It is immediately routed into an Online Feedback Loop:

$$\tau_{\text{eval}} \longrightarrow \text{Online Feedback} \longrightarrow \text{Storage Layer Update} \quad (9)$$

The execution outcomes modify ϕ_{succ} or ϕ_{fail} in real time, triggering automatic updates to confidence metrics and lifecycle states. This interaction closes the loop, realizing a fully adaptive, circular reasoning ecosystem.

4 Experiments

4.1 Dataset

To comprehensively validate the effectiveness and generalizability of the proposed framework across diverse application scenarios, we evaluate XSKILL on five benchmarks spanning three distinct domains. The first domain corresponds to visual agentic tool use, with a particular emphasis on visual reasoning and multi-tool manipulation. Specifically, we adopt VisualToolBench (Guo et al., 2025) and TIRBench (Li et al., 2025a), both of which require agents to process visual inputs and invoke appropriate tools to perform fine-grained image analysis. The second domain focuses on multimodal search, which involves information retrieval and web browsing over heterogeneous textual and visual sources. To this end, we employ MMSearch-Plus (Tao et al., 2025) and MMBrowseComp (Li et al., 2025b), thereby assessing agents' capability

Benchmark	Average@4		Pass@4	
	XSkill	πSkill	XSkill	πSkill
VisualToolBench	9.11	**11.33**$^{+2.22}$	21.03	**25.23**$^{+4.20}$
TIR-Bench	28.12	**29.13**$^{+1.01}$	58.50	**60.00**$^{+1.50}$
MMSearch-Plus	4.62	**6.00**$^{+1.38}$	11.00	**13.50**$^{+2.50}$
AgentVista	7.80	**9.63**$^{+1.83}$	15.60	**20.18**$^{+4.58}$
Avg	12.41	**14.02**$^{+1.61}$	26.53	**29.73**$^{+3.20}$

Table 1: Main comparison results of XSkill and πSkill on Qwen2.5-VL-7B. We report Average@4 and Pass@4 over four independent rollouts on VisualToolBench, TIR-Bench, MMSearch-Plus, and AgentVista. Superscript values indicate the absolute improvement of πSkill over XSkill.

to search, integrate, and reason over multimodal information. Finally, we include AgentVista (Su et al., 2026), an ultra-challenging comprehensive benchmark that jointly evaluates tool-use competence and complex search ability. For each benchmark, we construct the training set by randomly sampling 100 tasks for experience accumulation, while reserving the remaining tasks for evaluation.

4.2 Implementation Details

We evaluate our framework using Qwen2.5-VL-7B-Instruct (Bai et al., 2025). All experiments are conducted on four RTX PRO 6000 GPUs (NVIDIA, 2025). In the main experiments, Qwen2.5-VL-7B-Instruct accumulate experiences and skills from their own reasoning trajectories. For indexing, we use bge-small-en-v1.5 (Xiao et al., 2023). During inference, we set the retrieval top-k to 3. The generation parameters are set to temperature $T = 0.6$, top-$p = 1.0$, and the maximum number of turns to 20.

4.3 Evaluation Metrics

Success rate (SR) is employed as the primary evaluation indicator throughout our experiments. We perform $N = 4$ rollouts per task to comprehensively evaluate agent performance. We mainly report two metrics: (1) Average@N ($N = 4$), calculated as the mean success rate across all rollouts to reflect the agent's stability and reliability; (2) Pass@N ($N = 4$), defined as the percentage of tasks achieving success in no less than one rollout, which reveals the maximum potential of the agent.

4.4 Baselines

We compare πSkill with XSkill, a strong continual learning framework for multimodal agents. XSkill introduces a dual-stream knowledge accumulation mechanism that stores task-level skills and action-level experiences in an external knowledge base. During accumulation, XSkill extracts reusable skills and experiences from multi-path rollouts through visually grounded summarization and cross-rollout critique. During inference, it retrieves relevant knowledge according to the current multimodal task and injects the adapted knowledge into the agent prompt to improve tool use and reasoning performance.

Although XSkill provides an effective foundation for training-free continual improvement, it still has two limitations that motivate our design. First, its experience generation mainly depends on cross-rollout comparison, which focuses on coarse trajectory-level differences and may overlook fine-grained step-level causal factors within a single trajectory. Second, its memory management is relatively static, making it difficult to dynamically evaluate, update, or suppress experiences according to their later usage outcomes. Therefore, we use XSkill as the main baseline to evaluate whether the proposed Circular Step-level Knowledge Distillation (CSD) and Confidence-aware Lifecycle Memory Updating (CMU) can further improve multimodal agent performance.

For fair comparison, both XSkill and πSkill use the same backbone models, tool settings, training samples for knowledge accumulation, and evaluation protocol. We evaluate both methods on four representative multimodal agent benchmarks, including VisualToolBench, TIR-Bench, MMSearch-Plus, and AgentVista. Following XSkill, we report Average@4 and Pass@4 over four independent rollouts. Average@4 measures the average success rate across all rollouts, reflecting rollout-level reliability, while Pass@4 measures whether at least one rollout succeeds, reflecting the upper-bound problem-solving capability under multiple attempts.

4.5 Main Results

Table 1 presents the main comparison between XSkill and πSkill on Qwen2.5-VL-7B. Across four multimodal agent benchmarks, πSkill consistently outperforms XSkill, improving the average Average@4 from 12.41 to 14.02 and the average Pass@4 from 26.53 to 29.73. These results show that the proposed CSD and CMU modules enhance both rollout-level reliability and the chance of obtaining at least one successful trajectory. The

improvements are observed on VisualToolBench, TIR-Bench, MMSearch-Plus, and AgentVista, suggesting that fine-grained step-level knowledge distillation and confidence-aware lifecycle memory updating provide stable benefits for multimodal agent reasoning with Qwen2.5-VL-7B.

4.6 Ablation Study

To validate the effectiveness of the proposed components in πSkill, we conduct ablation studies on two core modules: (1) Circular Step-level Knowledge Distillation (CSD), and (2) Confidence-aware Lifecycle Memory Updating (CMU). CSD is implemented in the experience critique module, where single trajectories are decomposed into successful and failed step transitions to distill fine-grained tactical knowledge. CMU is implemented in the experience manager and version controller, where each experience is equipped with confidence scores, usage statistics, lifecycle states, and version records for online refinement. We compare four configurations:

Configuration 1: XSkill. This variant follows the original XSkill pipeline without our proposed CSD and CMU modules. It relies mainly on multi-rollout comparison and static experience storage. Although it can accumulate reusable knowledge from past trajectories, it lacks step-level intra-trajectory analysis and cannot dynamically adjust the reliability of stored experiences according to later usage outcomes.

Configuration 2: XSkill+CSD. This variant adds Circular Step-level Knowledge Distillation while removing Confidence-aware Lifecycle Memory Updating. In our implementation, CSD analyzes each rollout internally by separating successful steps, failed steps, final task success, and error reasons. This enables the system to extract localized knowledge from single trajectories rather than depending only on coarse cross-rollout comparison. However, without CMU, the generated experiences are still stored in a relatively static manner and cannot be continuously updated according to their downstream effectiveness.

Configuration 3: XSkill+CMU. This variant keeps the original knowledge distillation process but introduces Confidence-aware Lifecycle Memory Updating. Specifically, each experience is stored with confidence, success count, failure count, status, version, and parent-version metadata. During online use, successful experiences are reinforced, failed experiences are penalized, and persistently unreliable experiences can be pruned, archived, or marked for review. This tests whether lifecycle-aware memory management alone can improve agent performance without more fine-grained step-level knowledge extraction.

Configuration 4: πSkill. This is our full framework, which integrates both CSD and CMU. CSD improves the density and granularity of extracted knowledge by mining step-level success–failure patterns from individual trajectories, while CMU ensures that the resulting knowledge base evolves according to real usage feedback. Together, these two modules form a closed loop from fine-grained knowledge generation to confidence-aware online memory updating.

Table 2 shows that the full πSkill framework consistently achieves the best results across all four benchmarks. Compared with XSkill, πSkill improves the overall Average@4 from 12.41 to 14.02 and the overall Pass@4 from 26.53 to 29.73, indicating that our method improves both rollout-level reliability and the probability of obtaining at least one successful trajectory.

The single-module variants reveal different but incomplete effects. **XSkill+CSD** improves VisualToolBench and MMSearch-Plus, suggesting that step-level distillation can extract useful localized knowledge for visual tool-use and multimodal search tasks. However, its performance drops on TIR-Bench, showing that fine-grained knowledge extraction alone may introduce noisy or unstable experiences when the memory system lacks confidence-aware filtering and lifecycle control. **XSkill+CMU** improves VisualToolBench and AgentVista, demonstrating that dynamic confidence tracking and versioned memory management help preserve more reliable experiences. Nevertheless, without CSD, the system still depends on coarser knowledge generated from the original critique process, limiting its ability to capture step-level causal factors behind success and failure.

In contrast, **πSkill** combines the strengths of both modules. CSD provides high-density step-level knowledge by identifying useful actions and failure patterns within individual rollouts, while CMU prevents the memory bank from accumulating unreliable experiences through confidence adjustment, pruning, archiving, and version updates. This synergy leads to consistent improvements on all datasets, including VisualToolBench, TIR-Bench, MMSearch-Plus, and AgentVista. These results verify that πSkill's performance gain comes not

Methods	VisualToolBench		TIR-Bench		MMSearch-Plus		AgentVista		Avg	
	Average@4	Pass@4	Average@4	Pass@4	Average@4	Pass@4	Average@4	Pass@4	Average@4	Pass@4
XSkill	9.11	21.03	28.12	58.50	4.62	11.00	7.80	15.60	12.41	26.53
XSkill+CSD	$10.28^{+1.17}$	$22.43^{+1.40}$	$25.38^{-2.74}$	$51.50^{-7.00}$	$5.00^{+0.38}$	$11.50^{+0.50}$	$7.80^{+0.00}$	$16.51^{+0.91}$	$12.12^{-0.29}$	$25.49^{-1.04}$
XSkill+CMU	$10.63^{+1.52}$	$24.77^{+3.74}$	$25.75^{-2.37}$	$51.50^{-7.00}$	$3.87^{-0.75}$	$10.00^{-1.00}$	$8.72^{+0.92}$	$19.27^{+3.67}$	$12.24^{-0.17}$	$26.39^{-0.14}$
πSkill	$\mathbf{11.33^{+2.22}}$	$\mathbf{25.23^{+4.20}}$	$\mathbf{29.13^{+1.01}}$	$\mathbf{60.00^{+1.50}}$	$\mathbf{6.00^{+1.38}}$	$\mathbf{13.50^{+2.50}}$	$\mathbf{9.63^{+1.83}}$	$\mathbf{20.18^{+4.58}}$	$\mathbf{14.02^{+1.61}}$	$\mathbf{29.73^{+3.20}}$

Table 2: Ablation study results on Qwen2.5-VL-7B. **CSD**: Circular Step-level Knowledge Distillation, which extracts fine-grained knowledge from successful and failed step transitions within a single trajectory. **CMU**: Confidence-aware Lifecycle Memory Updating, which dynamically updates experience confidence, usage statistics, lifecycle status, and version history according to online feedback. Average@4 measures the average success rate over four independent rollouts, while Pass@4 measures whether at least one rollout succeeds. Superscript values indicate the absolute performance difference compared with XSkill. The results show that using either CSD or CMU alone brings partial and unstable improvements, whereas πSkill achieves the best performance across all benchmarks.

from a single isolated component, but from the closed-loop integration of fine-grained knowledge generation and confidence-aware online memory evolution.

5 Conclusion

In this work, we propose πSkill, a continual learning framework for multimodal agents that extends XSkill with fine-grained knowledge extraction and confidence-aware memory evolution. Specifically, Circular Step-level Knowledge Distillation (CSD) extracts high-density experiences from successful and failed step transitions within individual trajectories, while Confidence-aware Lifecycle Memory Updating (CMU) dynamically updates experience confidence, status, and version history according to online feedback.

Experiments on multiple multimodal agent benchmarks show that πSkill consistently outperforms XSkill across different backbone models in both Average@4 and Pass@4. Ablation results further verify that CSD improves the granularity of experience generation, while CMU enhances the reliability of the memory bank by reinforcing useful experiences and suppressing noisy ones. Overall, πSkill provides a more adaptive and reliable training-free mechanism for continual improvement in multimodal agents. Future work will explore its application to longer-horizon interactive tasks and broader cross-model knowledge transfer.

Limitations

As an early implementation of an experience- and skill-driven multimodal agent framework, XSkill has limitations regarding reproducibility and deployment generality. First, the current codebase relies on multiple external APIs, model endpoints, and third-party tool services, which may introduce instability caused by network latency, service availability, and backend-specific function-calling behaviors; thus, future work should provide stronger configuration validation, more standardized local backends, and more deterministic execution support. Second, although the framework supports several tools and benchmark-style inputs, we have not yet fully optimized it for heterogeneous real-world environments with different tool chains, data formats, and resource constraints. Given its modular design and training-free memory mechanism, our codebase remains a promising foundation for scalable multimodal agents, a potential we plan to further validate through broader deployment and more comprehensive engineering tests.

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, and 8 others. 2025. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Zheng Chu, Xiao Wang, Jack Hong, Huiming Fan, Yuqi Huang, Yue Yang, Guohai Xu, Chenxiao Zhao, Cheng Xiang, Shengchao Hu, and 1 others. 2026. Redsearcher: A scalable and cost-efficient framework for long-horizon search agents. *arXiv preprint arXiv:2602.14234*.

Xinyu Geng, Peng Xia, Zhen Zhang, Xinyu Wang, Qiuchen Wang, Ruixue Ding, Chenxi Wang, Jialong Wu, Yida Zhao, Kuan Li, and 1 others. 2025. Webwatcher: Breaking new frontier of vision-language deep research agent. *arXiv preprint arXiv:2508.05748*.

Xingang Guo, Utkarsh Tyagi, Advait Gosai, Paula Ver-

gara, Jayeon Park, Ernesto Gabriel Hernández Montoya, Chen Bo Calvin Zhang, Bin Hu, Yunzhong He, Bing Liu, and 1 others. 2025. Beyond seeing: Evaluating multimodal llms on tool-enabled image perception, transformation, and reasoning. *arXiv preprint arXiv:2510.12712*.

Zexue He, Yu Wang, Churan Zhi, Yuanzhe Hu, Tzu-Ping Chen, Lang Yin, Ze Chen, Tong Arthur Wu, Siru Ouyang, Zihan Wang, and 1 others. 2026. Memoryarena: Benchmarking agent memory in interdependent multi-session agentic tasks. *arXiv preprint arXiv:2602.16313*.

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

Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, and 1 others. 2025. Memory in the age of ai agents. *arXiv preprint arXiv:2512.13564*.

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

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

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

Shilong Li, Xingyuan Bu, Wenjie Wang, Jiaheng Liu, Jun Dong, Haoyang He, Hao Lu, Haozhe Zhang, Chenchen Jing, Zhen Li, and 1 others. 2025b. Mm-browsecomp: A comprehensive benchmark for multimodal browsing agents. *arXiv preprint arXiv:2508.13186*.

Junming Liu, Yifei Sun, Weihua Cheng, Haodong Lei, Yirong Chen, Licheng Wen, Xuemeng Yang, Daocheng Fu, Pinlong Cai, Nianchen Deng, and 1 others. 2025. Memverse: Multimodal memory for lifelong learning agents. *arXiv preprint arXiv:2512.03627*.

NVIDIA. 2025. Nvidia rtx pro 6000 blackwell workstation edition. https://www.nvidia.com/en-us/products/workstations/professional-desktop-gpus/rtx-pro-6000/.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, and 1 others. 2025. Reasoningbank: Scaling agent self-evolving with reasoning memory. *arXiv preprint arXiv:2509.25140*.

Zhaochen Su, Jincheng Gao, Hangyu Guo, Zhenhua Liu, Lueyang Zhang, Xinyu Geng, Shijue Huang, Peng Xia, Guanyu Jiang, Cheng Wang, and 1 others. 2026. Agentvista: Evaluating multimodal agents in ultra-challenging realistic visual scenarios. *arXiv preprint arXiv:2602.23166*.

Mirac Suzgun, Mert Yuksekgonul, Federico Bianchi, Dan Jurafsky, and James Zou. 2025. Dynamic cheatsheet: Test-time learning with adaptive memory, 2025. *URL https://arxiv. org/abs/2504.07952*.

Mirac Suzgun, Mert Yuksekgonul, Federico Bianchi, Dan Jurafsky, and James Zou. 2026. 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)*, pages 7080–7106.

Xiangru Tang, Tianrui Qin, Tianhao Peng, Ziyang Zhou, Daniel Shao, Tingting Du, Xinming Wei, Peng Xia, Fang Wu, He Zhu, and 1 others. 2025. Agent kb: Leveraging cross-domain experience for agentic problem solving. *arXiv preprint arXiv:2507.06229*.

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

Shijian Wang, Jiarui Jin, Runhao Fu, Zexuan Yan, Xingjian Wang, Mengkang Hu, Eric Wang, Xiaoxi Li, Kangning Zhang, Li Yao, and 1 others. 2026. Museagent: A multimodal reasoning agent with stateful experiences. *arXiv preprint arXiv:2603.27813*.

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

Rong Wu, Xiaoman Wang, Jianbiao Mei, Pinlong Cai, Daocheng Fu, and 1 others. 2025. Evolver: Self-evolving llm agents through an experience-driven lifecycle. *arXiv preprint arXiv:2510.16079*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Yuxi Xie, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy Lillicrap, and Kenji Kawaguchi. 2024. Monte carlo tree search boosts reasoning via iterative preference learning. *arXiv preprint arXiv:2405.00451*.

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

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