Coalition-Aware Skill Reliability for Self-Evolving Agents

Qiyan Zhao[1,2], Xiaofeng Zhang[2,3], Bo Liu[1], Minda Chen[2], Wei Xiong[2], Jingyang Chen[2], Guanting Ye[5], Wenhao Yu[5], Xiaosong Yuan[3], Shijie Han[4], Da-Han Wang[1], Jianmin Ji[5], Fei Huang[2], Xu-Yao Zhang[1,†]

[1]CASIA [2]LongShine AI Lab [3]SJTU [4]HKUST [5]USTC

[†]Corresponding author

Abstract

Agent skills, structured artifacts distilled from interaction trajectories and dynamically reused from skill banks, have become a central mechanism for enabling large language model (LLM)-based self-evolving agents to learn from past experience. Yet existing work has largely focused on the operational aspects of skills, such as acquisition, evolution, and retrieval, while leaving a more fundamental reliability question unresolved: *Do accumulated skills in an agent's skill bank actually make positive mechanistic contributions?* We investigate this question through systematic skill-bank audits across alternative bank compositions and deployment domains, measuring the resulting changes in agent behavior. These audits reveal two recurring reliability failures: **coalition pollution**, where bank-level gains conceal negative coalition-level skill contributions, and **cross-domain utility reversal**, where source-beneficial skills reverse their effects after transfer. These findings motivate two reliability interventions: coalition-aware skill selection during skill accumulation and label-free skill masking after transfer. **Coalition-Aware Skill Selection (CASS)** selects more reliable candidate skills for the current bank using sampled Shapley marginals. **Unsupervised Skill-Masked Coalition Optimizer (u-SMCO)** masks transferred skills whose exclusion improves retrieval quality on unlabeled target-domain data. Agentic experiments on LoCoMo, LongMemEval, HotpotQA, and ALFWorld show that CASS and u-SMCO consistently improve task performance and cross-domain generalization over strong skill-based self-evolving agent baselines. Beyond accuracy, coalition-conditioned reliability modeling reduces sensitivity to noisy outcome-reward fluctuations during reinforcement learning and exposes the limits of isolation-based skill evaluation. These results establish skill reliability as a joint property of skills, banks, and deployment domains, with skill coalitions serving as the central unit of reliable skill evolution. Our audit toolkit, method code, and trained checkpoints will be released.

1 Introduction

Large language model (LLM)-based agents (Wei et al. 2026; Huang et al. 2026; Wang et al. 2023) are increasingly capable of self-evolving by learning from both successful and failed interactions. Recent works (Xia et al. 2026; Zhang et al. 2026b; Yang et al. 2026; Shi et al. 2026; Ouyang et al. 2026) advance this paradigm by distilling raw interaction trajectories into *skills* (Xu and Yan 2026): high-level, reusable, structured artifacts that abstract the essential knowledge for solving tasks. Once accumulated into a persistent skill bank, these artifacts are retrieved and instantiated at inference time without additional parameter updates (Berthon, Astorga, and van der Schaar 2026; Li et al. 2026c; Xu et al. 2026b).

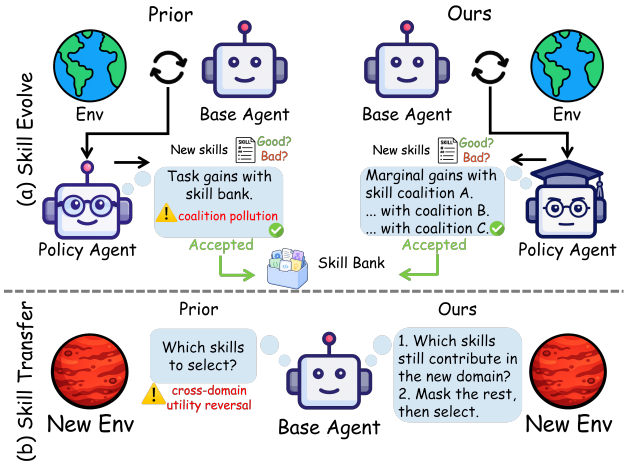

Figure 1: **Motivation.** (a) Existing self-evolving agents overlook coalition pollution during skill evolution, admitting unreliable skills with negative coalition-level contributions. (b) During cross-domain transfer, prior methods ignore cross-domain utility reversal, allowing skills that become unreliable in the target domain to influence subsequent decisions. These two mechanistic failure modes motivate our work.

Most existing work focuses on operations along the skill lifecycle, particularly skill acquisition (Qiu et al. 2026; Ni et al. 2026), evolution (Zhang et al. 2026a; Shen et al. 2026b), and retrieval (Xia et al. 2026; Su et al. 2026). Other studies have investigated how individual skills can be composed (Chen et al. 2026a; Li et al. 2026b) and evaluated their generalization across tasks and domains (Huang et al. 2026; Li et al. 2026c). Despite these practical successes in expanding agent capabilities, a more fundamental reliability question remains largely unexplored: ***Do the skills accumulated in an agent's skill bank actually make positive mechanistic contributions when invoked?***

To this end, we conduct a systematic Skill Mechanistic Reliability Audit (SMRA) on representative skill-based self-evolving agents (Zhang et al. 2026b). Our audit covers two lifecycle stages: skill evolution and cross-domain transfer

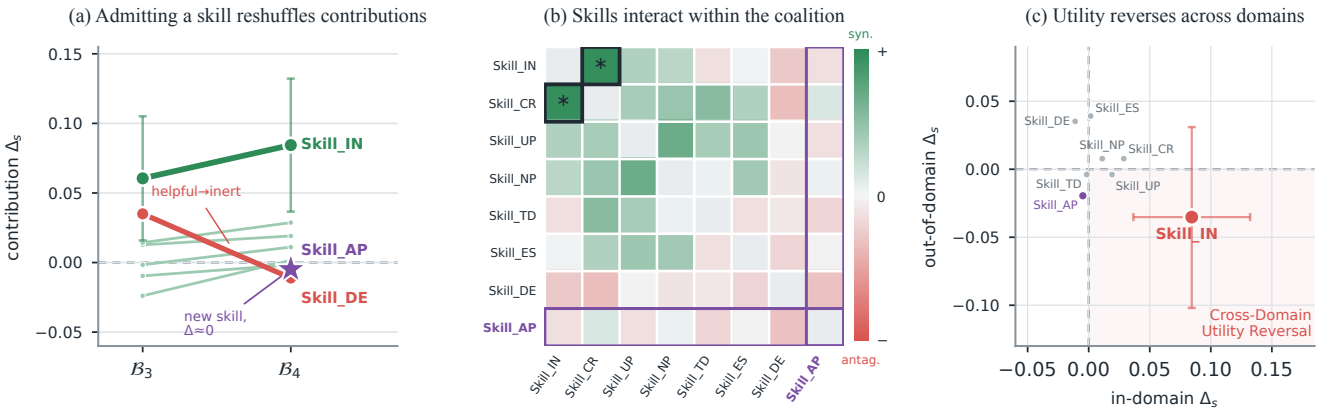

Figure 2: **Skill reliability is not intrinsic.** We audit the same skills across bank compositions and deployment domains (skills relabeled `Skill_XX`). (a) *Acceptance is not contribution:* the gate admits `Skill_AP` (purple star), yet it contributes ≈0 while the incumbent `Skill_IN` carries the bank. (b) *Contribution depends on partners:* the interaction matrix reveals synergies and antagonisms, strongest between `Skill_IN` and `Skill_CR` (boxed). (c) *Same bank, different verdict:* `Skill_IN` helps at the source (LoCoMo) and reverses at the target (HotpotQA-50).

(Figure 1). Both stages audit skill reliability by systematically varying the skill-bank composition. The evolution stage compares bank states before and after new skills are accepted (Figure 2a-b); the transfer stage compares in-domain versus out-of-domain deployment on the accumulated bank (Figure 2c). This audit reveals two key reliability failures:

Coalition Pollution. During skill evolution, a newly extracted candidate is accepted into the skill bank if its inclusion improves aggregate task performance. Yet this gain reveals neither which skill produces the value nor how skills interact within the bank. Comparing the audit results before and after acceptance (Figure 2a-b), we uncover an inversion: the admitted skill shows near-zero contribution while an incumbent carries the bank, and existing marginals shift with the composition. The gate thus retains skills whose mechanistic reliability it never established, because a skill's contribution is conditioned on its coalition rather than intrinsic. We term this phenomenon coalition pollution.

Cross-Domain Utility Reversal. During cross-domain transfer, agents carry the entire accumulated skill bank into the target domain and dynamically retrieve skills for downstream tasks. This practice implicitly assumes that skills useful in the source domain remain useful after transfer. Comparing the audit results across in-domain and out-of-domain deployment (Figure 2c), we find this assumption fails: some skills beneficial in the source domain reverse their effect at the target and degrade performance. We term this failure mode cross-domain utility reversal.

Self-evolving agents' skill banks contain two reliability failure modes: bank-level gains conceal coalition-level harm during evolution, and source-useful skills reverse to harmful under transfer.

Based on these findings, we introduce two interventions, one for each failure mode.

Coalition-Aware Skill Selection (CASS) addresses coalition pollution during skill evolution. Instead of relying on aggregate reward alone, CASS augments the acceptance criterion with a coalition-aware signal. When evaluating a candidate, it uses Monte Carlo sampling to construct diverse skill coalitions from the resulting bank and aggregates their Shapley marginal contributions to estimate coalition-conditioned reliability. This estimate is then combined with aggregate reward to decide acceptance.

Unsupervised Skill-Masked Coalition Optimizer (u-SMCO) addresses cross-domain utility reversal after transfer. Using unlabeled queries from the target domain, u-SMCO scores each skill by its retrieval-quality contribution and greedily masks skills whose removal improves retrieval. Because no task labels are required, u-SMCO applies at deployment without target-domain supervision.

Experiments on LoCoMo, LongMemEval, HotpotQA, and ALFWorld show that CASS and u-SMCO consistently improve task performance and cross-domain generalization. Further analysis shows that our coalition-aware signals mitigate reward noise during training; we further prove that outcome-only gates are structurally blind to the coalition-level interactions these failures rest on. Together, these results reveal a broader principle: skill reliability is a joint property of the skill, the surrounding skill bank, and the deployment domain, not an intrinsic attribute of individual skills.

Our contributions can be summarized as follows.

- We introduce SMRA and identify two reliability failure modes of self-evolving skills: coalition pollution during evolution and cross-domain utility reversal after transfer.
- We propose CASS, a coalition-aware method that selects more reliable candidate skills for the bank by augmenting aggregate reward with sampled Shapley marginals.
- We propose u-SMCO, a label-free method that masks skills whose utility reverses after transfer using only unlabeled target-domain queries.
- We prove that outcome-only gates are structurally blind to the coalition-level interactions that drive both failure modes, motivating our two interventions.
- Across four benchmarks, CASS and u-SMCO consistently improve task performance and cross-domain generalization over strong self-evolving baselines.

2 Related Work

2.1 Self-Evolving LLM Agents

Large language models (LLMs) (Huang et al. 2024; Yang et al. 2025; Agarwal et al. 2025) have demonstrated strong reasoning capabilities across a wide range of complex tasks (Xu et al. 2026b; Maharana et al. 2024; Wu et al. 2025; Yang et al. 2018; Shridhar et al. 2020), accelerating the development of autonomous agent systems (Wei et al. 2026; Wang et al. 2023). Some studies have further explored in-context learning to enhance agent reasoning (Wu et al. 2023; Zhao et al. 2024). However, these agents typically treat each interaction as an isolated episode and approach every new task from scratch, without leveraging prior experience. This paradigm limits their ability to adapt to increasingly complex or long-horizon tasks in dynamic, open-ended environments.

To enable LLM-based agents to learn from past interactions, memory-based self-evolving agents (Xu et al. 2026a; Wang and Chen 2025; Yu et al. 2024; Kang et al. 2025) store sampled interaction trajectories in external databases as reusable experience. However, raw trajectories often contain substantial redundancy and noise (Zhao et al. 2024; Chhikara et al. 2025; Mi et al. 2026), motivating the development of skill-based self-evolving agents (Zhang, Fu, and Yan 2025; Ouyang et al. 2025; Wei et al. 2025) that distill historical trajectories into compact, reusable behavioral primitives, namely skills (Tu et al. 2026; Zhu et al. 2026). In parallel, advances in reinforcement learning (Chen et al. 2026b; Yu et al. 2026; Wang et al. 2026) have provided stronger supervisory signals for agent self-evolution.

2.2 Agent Skills

With the emergence of agent scaffolds such as Claude Code and OpenClaw (Xu and Yan 2026; Li et al. 2026a; He et al. 2026), *Agent Skills* have become a structured mechanism for encoding reusable decision-making strategies (Shen et al. 2026a; Liang et al. 2026). These behavioral primitives can be retrieved at inference time without updating model parameters, allowing agents to reuse knowledge distilled from both successful and failed interaction trajectories (Qiu et al. 2026; Mi et al. 2026).

Existing work spans the full skill lifecycle (Xu and Yan 2026; Vishe et al. 2026): Trace2Skill (Ni et al. 2026) extracts skills from execution logs, MemP (Fang et al. 2026) formalizes construction, retrieval, and updating as executable programs, MemSkill (Zhang et al. 2026b) develops skills for memory operations, CoEvoSkills (Zhang et al. 2026a) refines skills through co-evolutionary validation, SkillRL (Xia et al. 2026) and other reinforcement-learning approaches (Alzubi et al. 2026; Wang et al. 2026) refine skill banks with reward signals, Skill0 (Lu et al. 2026) internalizes acquired skills into model parameters, and SkillGraph (Li et al. 2026b) models inter-skill dependencies with graph structures. Despite these advances, existing methods largely emphasize the operational aspects of skills, leaving a fundamental reliability question unresolved: *Do accumulated skills in an agent's skill bank make positive mechanistic contributions when invoked?* Our work addresses this question by examining skill reliability during evolution and cross-domain transfer.

2.3 Skill Reliability

The concurrent work most closely related to ours is SkillLens (Huang et al. 2026), which characterizes skill reliability from a utility-grounded perspective and shows that textually plausible skills can still have negative effects. Our work is complementary along three axes: we examine reliability through *causal* contributions to downstream performance rather than text-based utility estimation; we adopt a *coalition-aware* view in which reliability is a joint property of the skill, its surrounding bank, and the deployment domain, rather than an intrinsic per-skill attribute; and we go beyond diagnosis to propose two mechanistic interventions that address the failure modes we identify.

Our approach builds on two methodological lineages: coalition-level contribution attribution via the Shapley value (Shapley et al. 1953; Lundberg and Lee 2017) and its higher-order interaction extensions (Sundararajan and Najmi 2020; Grabisch et al. 2016); and model reliability under distribution shift (Iwasawa and Matsuo 2021; Wang et al. 2020), which shows that source-domain competence is a poor predictor of target-domain behavior.

3 Skill Mechanistic Reliability Audit

In this section, we begin with a brief overview of the baseline agent's skill-bank evolution mechanism. We then apply SMRA to trace skill contributions across bank compositions and deployment domains, revealing coalition pollution and cross-domain utility reversal. Finally, we analyze why these failures arise, motivating our method design.

3.1 Self-Evolution and SMRA Formulation

Consider a skill-based agent built on a base agent F. At evolution stage e, let $\mathcal{B}_e = \{s_1, \ldots, s_n\}$ denote its current skill bank, where each s_i is a reusable skill. Given the interaction history accumulated by F, a skill-evolution policy F' proposes a candidate skill s_{new}, yielding the candidate bank $\widetilde{\mathcal{B}}_{e+1} = \mathcal{B}_e \cup \{s_{\text{new}}\}$. Under an outcome-only gate, whether s_{new} is retained is determined by the bank-level gain

$$G_e = r_{\text{out}}(\widetilde{\mathcal{B}}_{e+1}) - r_{\text{out}}(\mathcal{B}_e), \tag{1}$$

where $r_{\text{out}}(\mathcal{B})$ is the aggregate outcome reward estimated online from recent training interactions with bank $\mathcal{B}$. If $G_e > 0$, the candidate is accepted and $\mathcal{B}_{e+1} = \widetilde{\mathcal{B}}_{e+1}$; otherwise, $\mathcal{B}_{e+1} = \mathcal{B}_e$. This gate captures only bank-level gains, not how skill contributions vary across bank compositions.

To test whether accumulated skills make positive mechanistic contributions, we introduce the *Skill Mechanistic Reliability Audit* (SMRA). SMRA adopts Shapley's marginal-contribution view (Shapley et al. 1953; Lundberg and Lee 2017): skills are players, and $V^D(\mathcal{C})$ gives the downstream value of any skill coalition $\mathcal{C} \subseteq \mathcal{B}$ on domain D. For each $s \in \mathcal{B}$, SMRA audits its full-bank marginal contribution through a knockout intervention:

$$\Delta_s^D(\mathcal{B}) = V^D(\mathcal{B}) - V^D(\mathcal{B} \setminus \{s\}). \tag{2}$$

Full-bank and knockout evaluations are paired on identical examples. Positive Δ_s^D means s supports the bank in D;

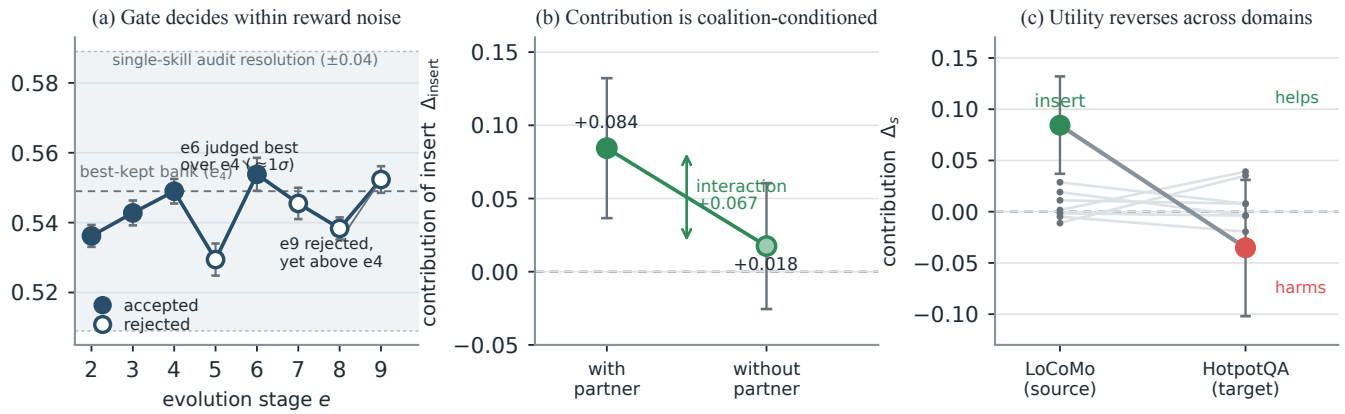

Figure 3: **Reliability failures revealed by SMRA.** (a) *Decisions at the noise floor:* accept/reject margins are commensurate with the ± 1 SE error bars; the shaded band marks the ± 0.04 single-skill audit resolution, an order of magnitude wider. (b) *Partner removed, marginal drops:* `insert` falls from $+0.084$ to $+0.018$ once `capture_contextual_relationships` is removed. (c) *Source benefit, target reversal:* `insert` helps on the source (LoCoMo), reverses on the target (HP-50).

negative values mean its knockout improves performance. To account for composition dependence, the Shapley value generalizes this full-bank marginal by taking a weighted average over every subset $\mathcal{C} \subseteq \mathcal{B} \setminus \{s\}$ of the other skills:

$$\phi_s^D(\mathcal{B}) = \sum_{\mathcal{C} \subseteq \mathcal{B} \setminus \{s\}} \frac{|\mathcal{C}|!(|\mathcal{B}| - |\mathcal{C}| - 1)!}{|\mathcal{B}|!} \cdot \left[V^D(\mathcal{C} \cup \{s\}) - V^D(\mathcal{C}) \right]. \tag{3}$$

Thus, $\phi_s^D(\mathcal{B})$ summarizes the expected marginal contribution of s across alternative bank compositions. Collectively, these values form a mechanistic reliability profile of $\mathcal{B}$. Repeating the audit across evolving banks and deployment domains exposes coalition pollution and cross-domain utility reversal, respectively.

3.2 Coalition Pollution

An outcome-only gate judges a candidate by a single scalar: it accepts when the bank's aggregate reward rises, $G_e > 0$. That scalar is an online, bank-level summary of the transition; it does not measure which skill in the resulting bank contributes. Our audit asks that separate question, isolating each skill by a knockout paired on identical held-out examples (Eq. 2). In our baseline agent, the gate admits $\mathcal{B}_3 \to \mathcal{B}_4$ on an online margin of only $+0.006$, which it cannot ascribe to any single skill; the paired audit of the resulting bank (Table 1) then finds the newly admitted `capture_activity_preferences` non-contributing ($\Delta = -0.005$), its pairwise interactions likewise negligible (Figure 2b), while `insert` is the skill actually carrying the bank ($\Delta = +0.084$). Nor is contribution an intrinsic property of a skill: the marginal of `insert` drops sharply once its coalition partner `capture_contextual_relationships` is removed (Figure 3(b)). Reliability is therefore a property of the coalition rather than the individual skill; we term the retention of non-contributing skills under such a gate *coalition pollution*.

The gate cannot prevent this. Figure 3(a) shows why: the accept/reject margins it acts on (0.006 to 0.02) are only one to a few times the per-stage standard error of its own reward estimate (error bars, ≈ 0.004), so statistically indistinguishable banks are ranked as decisively better or worse. The per-skill contributions that would justify these rankings are further out of reach: auditing a single skill on 314 paired queries resolves its contribution only to $\approx \pm 0.04$ (shaded band), an order of magnitude coarser than the margins being acted on. This is not merely a sampling limitation but a structural one.

Table 1: **Per-skill SMRA contributions** before ($\mathcal{B}_3$) and after ($\mathcal{B}_4$) the gate's acceptance (LoCoMo). *: 95% CI excludes zero; †: newly accepted.

Skill	$\Delta_s(\mathcal{B}_3)$	$\Delta_s(\mathcal{B}_4)$	Full name
`Skill_IN`	$+0.061^*$	$+0.084^*$	`insert`
`Skill_DE`	$+0.035$	-0.011	`delete`
`Skill_UP`	$+0.013$	$+0.019$	`update`
`Skill_NP`	-0.002	$+0.011$	`noop`
`Skill_TD`	-0.010	-0.002	`capture_temporal_details`
`Skill_CR`	$+0.014$	$+0.029$	`capture_contextual_relationships`
`Skill_ES`	-0.024	$+0.002$	`capture_entity_specifics`
`Skill_AP`†	—	-0.005	`capture_activity_preferences`

Theorem 1 (Outcome-only non-identifiability)**.** *For a gate observing only the endpoints $V(\mathcal{B}_e)$ and $V(\widetilde{\mathcal{B}}_{e+1})$, a positive gain $V(\widetilde{\mathcal{B}}_{e+1}) - V(\mathcal{B}_e) > 0$ does not identify whether $\widetilde{\mathcal{B}}_{e+1}$ contains a negatively contributing skill.*

In the audit's terms, a noise-free gate reads exactly one entry of the bank's reliability profile: G_e is the candidate's own marginal $\Delta_{s_{\text{new}}}(\widetilde{\mathcal{B}}_{e+1})$; every other member is left unconstrained, and Appendix A.1 constructs two banks with identical endpoints yet opposite signs for one member's contribution. CASS closes this gap by sampling coalition knockouts during evolution, an audit under alternative compositions.

3.3 Cross-Domain Utility Reversal

A skill selected in one domain is carried unchanged into another, implicitly assuming its usefulness transfers. It need not. Holding the bank fixed and varying only the deployment

domain, a skill exhibits *cross-domain utility reversal* when its contribution is positive at the source D_{src} yet negative at the target D_{tgt},

$$\Delta_s^{D_{\text{src}}}(\mathcal{B}) > 0 \quad \text{and} \quad \Delta_s^{D_{\text{tgt}}}(\mathcal{B}) < 0. \tag{4}$$

The clearest case is `insert`. On the source domain LoCoMo it is significantly beneficial ($\Delta = +0.084$, 95% CI excluding zero), yet this benefit does not survive transfer: on HotpotQA its point estimate reverses to $\Delta = -0.035$, though with a 95% CI that still includes zero (Figure 3(c)). We therefore read the target side as a directional reversal rather than significant harm. Because the bank is identical in both evaluations and only the deployment distribution changes, a contribution that is reliably positive at the source carries no such guarantee at the target.

The cause is that the value function is itself domain-dependent: the target distribution changes both where a skill is exercised and what its contributions are worth, so a source-domain contribution cannot be carried across domains as an invariant. A training-time gate sees only $V^{D_{\text{src}}}$ and has no signal fixing the sign of $\Delta_s^{D_{\text{tgt}}}$. Correcting cross-domain utility reversal therefore requires observing the target distribution; because target labels are often unavailable, u-SMCO relies instead on an unlabeled retrieval-quality signal, masking skills whose removal improves retrieval on target queries.

3.4 A Unified Reliability View

Coalition pollution and cross-domain utility reversal are two faces of one fact: skill reliability is not intrinsic. A skill's contribution depends on the bank it sits in and on the domain where it is deployed, so we write it as a contextual quantity $R(s; \mathcal{B}, D)$ rather than a fixed score $R(s)$. An outcome-only gate, which reads a single bank-level scalar at the source domain, is blind to both dependencies by construction. The two failures also mark where the missing signal can be restored: during evolution, CASS re-audits the candidate bank under sampled coalition knockouts before retaining it, supplying the composition dependence the gate lacks; after transfer, u-SMCO re-scores the transferred bank with a label-free, target-conditioned retrieval signal, supplying the domain dependence the gate never observed. Both apply the same principle at the two stages where context changes: reliability is judged on coalitions, never on skills in isolation.

Do accumulated skills actually contribute? The question has no skill-intrinsic answer: contribution depends on the surrounding coalition and the deployment domain, and bank-level success is evidence of neither.

4 Method

To address coalition pollution and cross-domain utility reversal, we propose two reliability interventions, Coalition-Aware Skill Selection (CASS) and Unsupervised Skill-Masked Coalition Optimizer (u-SMCO), respectively.

4.1 CASS: Coalition-Aware Skill Selection

Section 3.2 audited reliability after the fact; CASS embeds that audit into skill evolution itself. Whenever the gate chooses between candidate banks that differ in composition, the signal it needs is the composition-averaged contribution formalized by Eq. (3). Estimating that value to useful precision takes hundreds of permutation samples per skill (Appendix A.3), each rebuilding the agent's memory under the ablated composition and re-evaluating the bank, which is prohibitive at every gate decision. CASS therefore retains the coalition view but abandons per-skill attribution, estimating instead a bank-level coalition average by Monte Carlo sampling over a bounded number of coalitions, trading exhaustive evaluation for a fixed stochastic budget as in sampling-based planning (Kocsis and Szepesvári 2006; Grabisch et al. 2016).

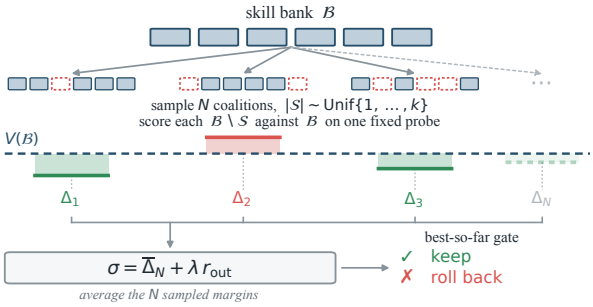

Figure 4: **One CASS gate decision.** Each sampled coalition is knocked out and scored against the full bank on one fixed probe, so every margin is a paired difference from $V(\mathcal{B})$; green marks removal that costs value, red that gains it. Algorithm A1 gives the full procedure.

Given a bank $\mathcal{B}$ and a coalition $\mathcal{S} \subseteq \mathcal{B}$, a subset of the skills it holds, CASS scores the bank in two steps. It first draws N distinct coalitions by Monte Carlo sampling over the bank (Figure 4): each draw takes a size $|\mathcal{S}|$ uniform on $\{1, \ldots, k\}$, then a coalition $\mathcal{S}$ uniform among the subsets of that size. It then evaluates every sampled coalition by its knockout margin,

$$\Delta_{\mathcal{S}}(\mathcal{B}) = \frac{V(\mathcal{B}) - V(\mathcal{B} \setminus \mathcal{S})}{|\mathcal{S}|}, \tag{5}$$

which charges the joint loss of removing $\mathcal{S}$ equally to its members. Sizes above one are what make the score more than a per-skill average: the joint loss departs from the sum of single-skill margins exactly when skills interact. CASS averages the N margins into $\hat{\bar{\Delta}}_N(\mathcal{B})$ and combines it with the online outcome reward into the scalar the gate ranks,

$$\sigma(\mathcal{B}) = \hat{\bar{\Delta}}_N(\mathcal{B}) + \lambda\, r_{\text{out}}(\mathcal{B}), \qquad \lambda = 0.2, \tag{6}$$

retaining $\widetilde{\mathcal{B}}_{e+1}$ iff its score exceeds every score seen so far. The rule is the baseline's best-so-far gate of Eq. (1) with only the ranked scalar changed.

4.2 u-SMCO: Unsupervised Bank Masking

CASS handles coalition pollution *during training*; cross-domain utility reversal instead calls for action at deployment. Given a trained bank $\mathcal{B}$ and unlabeled target queries Q_{tgt}, u-SMCO decides which skills to mask, so it needs a per-skill signal that is computable without labels and causally connected to downstream utility. Label-free is the easy half. The

	Conversational Benchmarks						Embodied Interactive Tasks				
	LoCoMo		LongMemEval		Avg.		ALF-Seen		ALF-Unseen		Avg.
Method	F1	L-J	F1	L-J	F1	L-J	SR	#Stps↓	SR	#Stps↓	SR
CoN	30.86	41.72	30.78	56.44	30.82	49.08	75.00	19.15	80.60	17.38	77.80
ReadAgent	28.63	38.25	24.48	42.62	26.56	40.44	62.86	26.14	71.64	22.88	67.25
MemoryBank	36.80	44.43	30.56	41.96	33.68	43.20	60.71	28.23	66.42	24.64	63.57
A-MEM	39.39	49.71	25.83	38.04	32.61	43.88	62.86	27.53	70.15	23.79	66.51
Mem0	25.48	34.58	30.25	46.81	27.87	40.70	74.29	19.77	81.34	17.15	77.82
MemoryOS	41.39	48.64	17.59	39.83	29.49	44.24	57.86	27.94	65.67	24.46	61.77
MemSkill†	39.45	54.78	30.02	53.47	34.74	54.13	74.01	19.97	80.09	19.04	77.05
Ours	**41.48**	**56.58**	**30.03**	**54.46**	**35.76**	**55.52**	**76.10**	**19.94**	**83.22**	**19.93**	**79.66**

Table 2: **Main results** (%). L-J is the LLM-as-Judge score; #Stps is the mean number of environment interactions (lower is better). † our reproduction from MemSkill's released project.

agent's own preference, the drop in selection-policy entropy when a skill is present, ranks `insert` highest on HP-50, the very skill SMRA exposes as negative at the target (Figure 2c): that policy was trained under the outcome gate and inherits its bias instead of tracking utility, echoing the collapse mode of entropy-minimization TTA (Iwasawa and Matsuo 2021). Appendix C.1 reports the ablation.

u-SMCO instead scores skills by retrieval quality on the target itself. For a set of skills $\mathcal{A} \subseteq \mathcal{B}$, let $\mathcal{M}(\mathcal{A})$ be the memory bank rebuilt over the target contexts using only $\mathcal{A}$, and let $\mathrm{Top}(q)$ collect the memories of $\mathcal{M}(\mathcal{A})$ nearest q in cosine similarity. Define

$$\mathrm{RQ}(\mathcal{A}) = \frac{1}{|Q_{\text{tgt}}|} \sum_{q \in Q_{\text{tgt}}} \mathrm{mean}_{m \in \mathrm{Top}(q)} \cos(e_q, e_m). \quad (7)$$

The mask score $\psi(s) = \mathrm{RQ}(\mathcal{B}) - \mathrm{RQ}(\mathcal{B} \setminus \{s\})$ is the audit's knockout comparison once more, scored by retrieval quality on the target rather than by task outcome. Unlike the entropy signal it is decoupled from the training reward channel: it asks whether a memory-writing skill improves the semantic locality of what it stores with respect to the queries the deployment distribution will ask.

Algorithm 1 applies this greedily, masking the lowest-scoring skill and stopping once no score falls below a threshold τ. Only unlabeled queries and their raw contexts are required. Rebuilding the memory dominates the cost at $O(K^2)$ rebuilds, 6–10 minutes per mask step in our setup and negligible beside training.

Algorithm 1 u-SMCO: Unsupervised Skill Masking

Require: bank $\mathcal{B}$, unlabeled queries Q_{tgt}, contexts C_{tgt}, stop threshold τ

kept $\leftarrow \mathcal{B}$
while |kept| > 1 **do**
  for each $s \in$ kept **do**
    Rebuild memory $\mathcal{M}(\text{kept} \setminus \{s\})$ over C_{tgt}
    $\psi(s) \leftarrow \mathrm{RQ}(\text{kept}) - \mathrm{RQ}(\text{kept} \setminus \{s\})$
  end for
  $s^\star \leftarrow \arg\min_s \psi(s)$
  if $\psi(s^\star) \geq \tau$ **then**
    break
  else
    kept $\leftarrow$ kept $\setminus \{s^\star\}$
  end if
end while
return kept

5 Experiments

In this section we evaluate both interventions in detail. Matched-protocol comparisons against strong self-evolving baselines establish their effectiveness, and ablations isolate each design choice.

5.1 Setup

Our base self-evolving agent framework is MemSkill (Zhang et al. 2026b), whose skill-extraction policy agent is Llama-3.3-70B-Instruct (Huang et al. 2024), matching the original; all experiments run on 8×H20 GPUs. We reproduce the baseline strictly under its original settings (Table 2). Our methods are compared under a fully matched protocol: CASS differs from MemSkill only in the gate, and u-SMCO is applied only before inference. Evaluation spans four domains: LoCoMo (Maharana et al. 2024), LongMemEval (Wu et al. 2025), HotpotQA (Yang et al. 2018) and ALFWorld (Shridhar et al. 2020). Unless noted, we report mean ± std over three seeds under a fixed judge.

5.2 Main Results

We compare our method against state-of-the-art agents: CoN (Yu et al. 2024), ReadAgent (Lee et al. 2024), MemoryBank (Zhong et al. 2024), A-MEM (Xu et al. 2026a), Mem0 (Chhikara et al. 2025), MemoryOS (Kang et al. 2025) and MemSkill (Zhang et al. 2026b); see Table 2. Our method delivers strong performance across the three benchmarks and consistently improves on the baseline. We attribute this to what CASS changes in MemSkill's gating mechanism: because a candidate is admitted only when its contribution survives the coalition audit, the skills that accumulate are mechanistically more reliable and cooperate at the coalition level with those already in the bank.

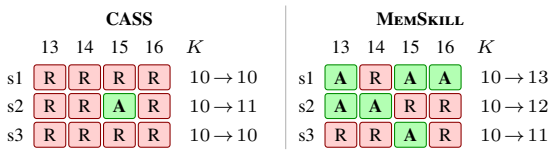

Figure 5: Gate decisions at the four outer epochs, and the resulting bank size. CASS accepts $1/12$, MemSkill $6/12$.

Benchmark	MemSkill e_{12}	MemSkill final	Δ
LoCoMo	0.5653	0.5303	-3.50
LME	0.5000	0.5248	$+2.48$
HP-50	0.3008	0.2695	-3.13
Avg.	0.4554	0.4415	-1.38

Table 3: K-controlled ablation of the gate's additions.

5.3 CASS Experiments

Across 12 candidate-skill proposals (4 gate decisions per seed, 3 seeds), CASS accepts 1 while MemSkill accepts 6 (Figure 5). A low acceptance rate is a virtue only if what it rejects deserved rejecting, so we audit what the outcome gate admitted. Holding K fixed, we compare the bank at e_{12} against the trained bank containing those additions (Table 3): they are collectively net-negative (-1.38 pp on average), although every one satisfied the outcome criterion when admitted, which is the endpoint ambiguity of Theorem 1 in practice. Figure 6 shows what the two gates admit.

5.4 u-SMCO Experiments

Table 4 reports u-SMCO on all six trained banks. `Skill_IN` is masked on 5/6 of them with no target label: the skill the audit finds carrying the bank at the source (Table 1) is the one the target probe rejects, which is cross-domain utility reversal detected label-free, and Figure 6 shows the skill itself. Masking improves every bank, but it improves the MemSkill banks twice as much as the CASS banks (7.17 vs. 3.59 pp on average): the pollution CASS keeps out at the gate is what u-SMCO must remove afterwards.

5.5 More Discussion

Reliability is a coalition-level property. Coalition pollution and cross-domain utility reversal share one root: reliability is a joint property of skill, bank and distribution, not an attribute of the skill. This is why judging a skill in isolation fails. The margins an outcome gate acts on lie below the resolution of a single-skill audit (Figure 3a), and Theorem 1 shows the limit is structural rather than statistical: no amount of outcome sampling separates a bank that improved from one that only appeared to. CASS therefore scores coalitions, not skills.

Why retrieval quality. At deployment the same question returns without labels, and the obvious signal is the agent's own preference. It fails. The drop in selection-policy entropy ranks `Skill_IN` as the bank's *most preferred* skill, precisely the one the target rejects, and anti-correlates with the label-based ranking ($\rho = -0.18$, Table 5); that policy was trained under the outcome gate and inherits its bias. Retrieval quality is measured on the target itself and is decoupled from the training reward, reaching $\rho = +0.76$.

Capture Temporal Context — accepted by the outcome gate

Purpose: Capture temporal context and relationships between events from the text chunk.

When to use: The text chunk mentions specific dates, time frames, or events related to each other.

How to apply: Attribute the temporal context to the correct events; ensure it is specific, relevant, and includes any causal relationships.

Capture Entity Attributes — accepted by CASS

Purpose: Capture detailed, factual information about entities from the text chunk, including attributes, relationships, and preferences.

When to use: The text chunk mentions specific details about an entity's attributes, relationships, or preferences.

How to apply: Attribute the detail to the correct entity; ensure the detail is specific, relevant, and factual.

Insert New Memory — masked by u-SMCO at the target

Purpose: Capture new, durable facts from the current text chunk that are missing in memory.

When to use: The text chunk introduces new facts, events, plans, or context worth storing.

How to apply: Compare against retrieved memories to avoid duplicates; split distinct facts into separate items.

Figure 6: **Case study.** A coalition-polluting skill accepted by the baseline gate, a coalition-aware skill accepted by CASS, and a utility-reversing skill masked by u-SMCO.

Bank	K	Masked skills	Δ HP-50 (pp)
CASS s1	10→9	`AP`	2.13
CASS s2	11→9	`IN`, `UP`	4.09
CASS s3	10→8	`ES`, `IN`	4.55
MemSkill s1	13→9	`ES`, `IN`, `TD`, `AP`	5.01
MemSkill s2	12→10	`IN`, `ES`	8.58
MemSkill s3	11→9	`TD`, `IN`	7.92

Table 4: u-SMCO on the six trained banks with an unlabeled 20-query target probe; Δ HP-50 is the LLM-Judge change from masking. Skill codes follow Table 1.

Cost. CASS adds eight coalition evaluations per outer epoch, about 5% of a 20-hour run, and u-SMCO runs once before deployment at $O(K^2)$ memory rebuilds. Neither adds any inference-time cost.

6 Conclusion

We recast skill reliability as a coalition-level property: a skill's contribution depends on the bank around it and on the domain it is deployed in. Under this view, outcome-only gates are provably blind to the two failures we identify, coalition pollution and cross-domain utility reversal. CASS addresses the first during training by scoring sampled coalitions rather than bank-level outcomes, and u-SMCO addresses the second after transfer using only unlabeled target queries. Across three seeds both improve the reliability-critical regimes they target. Two directions follow: adapting the coalition budget to how close candidate banks are, and extending the coalition view beyond textual skills.

References

Agarwal, S.; Ahmad, L.; Ai, J.; Altman, S.; Applebaum, A.; Arbus, E.; Arora, R. K.; Bai, Y.; Baker, B.; Bao, H.; et al. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.

Alzubi, S.; Provenzano, N.; Bingham, J.; Chen, W.; and Vu, T. 2026. Evoskill: Automated skill discovery for multi-agent systems. *arXiv preprint arXiv:2603.02766*.

Berthon, A.; Astorga, N.; and van der Schaar, M. 2026. Skill Neologisms: Towards Skill-based Continual Learning. In *Forty-third International Conference on Machine Learning*.

Chen, S.; Gai, J.; Zhou, R.; Zhang, J.; Zhu, T.; Li, J.; Wang, K.; Wang, Z.; Chen, Z.; Kaleb, K.; et al. 2026a. Skillcraft: Can LLM agents learn to use tools skillfully? *arXiv preprint arXiv:2603.00718*.

Chen, Y.; Wang, Y.; Zhang, Y.; Ye, Z.; Cai, Z.; Shi, Y.; Gu, Q.; Su, H.; Cai, X.; Wang, X.; et al. 2026b. Learning to self-verify makes language models better reasoners. *arXiv preprint arXiv:2602.07594*.

Chhikara, P.; Khant, D.; Aryan, S.; Singh, T.; and Yadav, D. 2025. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*.

Fang, R.; Liang, Y.; Wang, X.; Wu, J.; Qiao, S.; Xie, P.; Huang, F.; Chen, H.; and Zhang, N. 2026. Memp: Exploring agent procedural memory. In *Findings of the Association for Computational Linguistics: ACL 2026*, 17490–17502.

Grabisch, M.; et al. 2016. *Set functions, games and capacities in decision making*, volume 46. Springer.

He, C.; Zhou, X.; Wang, D.; Xu, H.; Liu, W.; and Miao, C. 2026. Openclaw as language infrastructure: A case-centered survey of a public agent ecosystem in the wild.

Huang, K. C.; Lakhotia, K.; Huang, K.; Chen, L.; Garg, L.; Lavender, A.; Silva, L.; Bell, L.; Zhang, L.; Guo, L.; et al. 2024. The llama 3 herd of models. *preprint*.

Huang, Z.; Xu, J.; Yang, Y.; Gong, Z.; Yang, Q.; Tian, M.; Wang, X.; Lv, C.; Gao, X.; Dai, Q.; et al. 2026. From raw experience to skill consumption: A systematic study of model-generated agent skills. *arXiv preprint arXiv:2605.23899*.

Iwasawa, Y.; and Matsuo, Y. 2021. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems*, 34: 2427–2440.

Kang, J.; Ji, M.; Zhao, Z.; and Bai, T. 2025. Memory os of ai agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 25972–25981.

Kocsis, L.; and Szepesvári, C. 2006. Bandit based monte-carlo planning. In *European conference on machine learning*, 282–293. Springer.

Lee, K.-H.; Chen, X.; Furuta, H.; Canny, J.; and Fischer, I. 2024. A human-inspired reading agent with gist memory of very long contexts. In *Proceedings of the 41st International Conference on Machine Learning*, 26396–26415.

Li, H.; Mu, C.; Chen, J.; Ren, S.; Cui, Z.; Zhang, Y.; Bai, L.; and Hu, S. 2026a. Organizing, orchestrating, and benchmarking agent skills at ecosystem scale. *arXiv preprint arXiv:2603.02176*.

Li, X.; Li, M.; Bao, K.; Ma, Y.; Wang, W.; Liu, D.; and Feng, F. 2026b. SkillGraph: Skill-Augmented Reinforcement Learning for Agents via Evolving Skill Graphs. *arXiv preprint arXiv:2605.12039*.

Li, X.; Liu, Y.; Chen, W.; You, B.; Di, Z.; He, Y.; Zheng, S.; Choe, K. W.; Sun, J.; Wang, S.; et al. 2026c. SkillsBench: Benchmarking how well agent skills work across diverse tasks. *arXiv preprint arXiv:2602.12670*.

Liang, Y.; Zhong, R.; Xu, H.; Jiang, C.; Zhong, Y.; Fang, R.; Gu, J.-C.; Deng, S.; Yao, Y.; Wang, M.; et al. 2026. Skillnet: Create, evaluate, and connect ai skills. *arXiv preprint arXiv:2603.04448*.

Lu, Z.; Yao, Z.; Wu, J.; Han, C.; Gu, Q.; Cai, X.; Lu, W.; Xiao, J.; Zhuang, Y.; and Shen, Y. 2026. Skill0: In-context agentic reinforcement learning for skill internalization. *arXiv preprint arXiv:2604.02268*.

Lundberg, S. M.; and Lee, S.-I. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Maharana, A.; Lee, D.-H.; Tulyakov, S.; Bansal, M.; Barbieri, F.; and Fang, Y. 2024. Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 13851–13870.

Mi, Q.; Ma, Z.; Yang, M.; Li, H.; Wang, Y.; Zhang, H.; and Wang, J. 2026. ProcMEM: Learning Reusable Procedural Memory from Experience via Non-Parametric PPO for LLM Agents. *arXiv preprint arXiv:2602.01869*.

Ni, J.; Liu, Y.; Liu, X.; Sun, Y.; Zhou, M.; Cheng, P.; Wang, D.; Zhao, E.; Jiang, X.; and Jiang, G. 2026. Trace2skill: Distill trajectory-local lessons into transferable agent skills. *arXiv preprint arXiv:2603.25158*.

Ouyang, S.; Yan, J.; Chen, Y.; Han, R.; Wang, Z.; Mishra, B. D.; Meng, R.; Li, C.-L.; Jiao, Y.; Zha, K.; et al. 2026. Skillos: Learning skill curation for self-evolving agents. *arXiv preprint arXiv:2605.06614*.

Ouyang, S.; Yan, J.; Hsu, I.; Chen, Y.; Jiang, K.; Wang, Z.; Han, R.; Le, L. T.; Daruki, S.; Tang, X.; et al. 2025. Reasoningbank: Scaling agent self-evolving with reasoning memory. *arXiv preprint arXiv:2509.25140*.

Qiu, L.; Gao, Z.; Chen, J.; Ye, Y.; Huang, W.; Xue, X.; Qiu, W.; and Tang, S. 2026. AutoRefine: From Trajectories to Reusable Expertise for Continual LLM Agent Refinement. *arXiv preprint arXiv:2601.22758*.

Shapley, L. S.; et al. 1953. A value for n-person games. In *Contributions to the Theory of Games*. Princeton University Press Princeton.

Shen, J.; Zhang, T.; Zhao, X.; and Cheng, H. 2026a. Dynamic skill lifecycle management for agentic reinforcement learning. *arXiv preprint arXiv:2605.10923*.

Shen, S.; Cheng, W.; Ma, M.; Turcan, A.; Zhang, M. J.; and Ma, J. 2026b. Skillfoundry: Building self-evolving agent skill libraries from heterogeneous scientific resources. *arXiv preprint arXiv:2604.03964*.

Shi, Y.; Chen, Y.; Lu, Z.; Miao, Y.; Liu, S.; Gu, Q.; Cai, X.; Wang, X.; and Zhang, A. 2026. Skill1: Unified evolution

of skill-augmented agents via reinforcement learning. *arXiv preprint arXiv:2605.06130*.

Shridhar, M.; Yuan, X.; Côté, M.-A.; Bisk, Y.; Trischler, A.; and Hausknecht, M. 2020. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*.

Su, W.; Long, J.; Ai, Q.; He, Q.; Tang, Y.; Wang, C.; Tu, Y.; Wang, Y.; and Liu, Y. 2026. Skill retrieval augmentation for agentic ai. *arXiv preprint arXiv:2604.24594*.

Sundararajan, M.; and Najmi, A. 2020. The many Shapley values for model explanation. In *International conference on machine learning*, 9269–9278. PMLR.

Tu, S.; Xu, C.; Zhang, Q.; Zhang, Y.; Lan, X.; Li, L.; Li, D.; and Zhao, D. 2026. Dynamic Dual-Granularity Skill Bank for Agentic RL. *arXiv preprint arXiv:2603.28716*.

Vishe, Y.; Surana, R.; Jiang, X.; Huang, Z.; Li, X.; Kuang, N. L.; Yu, T.; Rossi, R. A.; Shang, J.; McAuley, J.; et al. 2026. Skill-R1: Agent skill evolution via reinforcement learning. *arXiv preprint arXiv:2605.09359*.

Wang, D.; Shelhamer, E.; Liu, S.; Olshausen, B.; and Darrell, T. 2020. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*.

Wang, G.; Xie, Y.; Jiang, Y.; Mandlekar, A.; Xiao, C.; Zhu, Y.; Fan, L.; and Anandkumar, A. 2023. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*.

Wang, J.; Yan, Q.; Wang, Y.; Tian, Y.; Mishra, S. S.; Xu, Z.; Gandhi, M.; Xu, P.; and Cheong, L. L. 2026. Reinforcement learning for self-improving agent with skill library. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1529–1550.

Wang, Y.; and Chen, X. 2025. Mirix: Multi-agent memory system for llm-based agents. *arXiv preprint arXiv:2507.07957*.

Wei, T.; Li, T.-W.; Liu, Z.; Ning, X.; Yang, Z.; Zou, J.; Zeng, Z.; Qiu, R.; Lin, X.; Fu, D.; et al. 2026. Agentic reasoning for large language models. *arXiv preprint arXiv:2601.12538*.

Wei, T.; Sachdeva, N.; Coleman, B.; He, Z.; Bei, Y.; Ning, X.; Ai, M.; Li, Y.; He, J.; Chi, E. H.; et al. 2025. Evomemory: Benchmarking llm agent test-time learning with self-evolving memory. *arXiv preprint arXiv:2511.20857*.

Wu, D.; Wang, H.; Yu, W.; Zhang, Y.; Chang, K.-W.; and Yu, D. 2025. LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory. In *The Thirteenth International Conference on Learning Representations*.

Wu, Q.; Bansal, G.; Zhang, J.; Wu, Y.; Li, B.; Zhu, E.; Jiang, L.; Zhang, X.; Zhang, S.; Liu, J.; et al. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*.

Xia, P.; Chen, J.; Wang, H.; Liu, J.; Zeng, K.; Wang, Y.; Han, S.; Zhou, Y.; Zhao, X.; Chen, H.; et al. 2026. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning. *arXiv preprint arXiv:2602.08234*.

Xu, R.; and Yan, Y. 2026. Agent skills for large language models: Architecture, acquisition, security, and the path forward. *arXiv preprint arXiv:2602.12430*.

Xu, W.; Liang, Z.; Mei, K.; Gao, H.; Tan, J.; and Zhang, Y. 2026a. A-mem: Agentic memory for llm agents. *Advances in Neural Information Processing Systems*, 38: 17577–17604.

Xu, Z.; Feng, Y.; Dineen, J.; Shi, T.; Zhao, J.; and Zhou, B. 2026b. Skill Reuse as Compression in Agentic RL. *arXiv preprint arXiv:2605.31509*.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yang, Y.; Gong, Z.; Huang, W.; Yang, Q.; Zhou, Z.; Huang, Z.; Li, Y.; Gao, X.; Dai, Q.; Liu, B.; et al. 2026. Skillopt: Executive strategy for self-evolving agent skills. *arXiv preprint arXiv:2605.23904*.

Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W.; Salakhutdinov, R.; and Manning, C. D. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, 2369–2380.

Yu, Q.; Zhang, Z.; Zhu, R.; Yuan, Y.; Zuo, X.; Yue, Y.; Dai, W.; Fan, T.; Liu, G.; Liu, L.; et al. 2026. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38: 113222–113244.

Yu, W.; Zhang, H.; Pan, X.; Cao, P.; Ma, K.; Li, J.; Wang, H.; and Yu, D. 2024. Chain-of-note: Enhancing robustness in retrieval-augmented language models. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, 14672–14685.

Zhang, G.; Fu, M.; and Yan, S. 2025. Memgen: Weaving generative latent memory for self-evolving agents. *arXiv preprint arXiv:2509.24704*.

Zhang, H.; Fan, S.; Zou, H. P.; Chen, Y.; Wang, Z.; Zhou, J.; Li, C.; Huang, W.-C.; Yao, Y.; Zheng, K.; et al. 2026a. Coevoskills: Self-evolving agent skills via co-evolutionary verification. *arXiv preprint arXiv:2604.01687*.

Zhang, H.; Long, Q.; Bao, J.; Feng, T.; Zhang, W.; Yue, H.; and Wang, W. 2026b. MemSkill: Learning and Evolving Memory Skills for Self-Evolving Agents. *arXiv preprint arXiv:2602.02474*.

Zhao, A.; Huang, D.; Xu, Q.; Lin, M.; Liu, Y.-J.; and Huang, G. 2024. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 19632–19642.

Zhong, W.; Guo, L.; Gao, Q.; Ye, H.; and Wang, Y. 2024. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, 19724–19731.

Zhu, J.; Yu, J.; Zhao, Y.; Han, C.; Gu, Q.; Cai, X.; Li, X.; and Qian, W. 2026. Skill0. 5: Joint Skill Internalization and Utilization for Out-of-Distribution Generalization in Agentic Reinforcement Learning. *arXiv preprint arXiv:2605.28424*.