AECG: ASYMMETRIC EXPERIENCE CONSOLIDATION AND GOVERNANCE IN MULTI-AGENT SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM)-based multi-agent systems increasingly rely on memory to transform execution trajectories into reusable procedural knowledge. Yet repeated retrieval also makes memory errors persistent: memory pollution arises when outdated, weakly supported, or spuriously successful procedures become recurring components of future reasoning. Multi-agent execution introduces an additional structural risk. Scope collapse occurs when procedural knowledge escapes the coordination scope in which it was shown effective and is repeatedly reused at incompatible decision levels, allowing local errors to influence cascades of downstream decisions. Meanwhile, task-level failures provide ambiguous supervision because they rarely reveal which recalled knowledge was responsible. We introduce AECG, a framework for asymmetric experience consolidation and governance for multi-agent systems. AECG turns memory from static experience storage into a dynamic reliability-governance loop, preserving coordination scope and using multi-scale, confidence-aware reliability to detect degradation. It then combines degradation with downstream impact to prioritize high-risk knowledge under a bounded review budget, applies targeted interventions, and reactivates revised skills only after paired replay. Across three multi-agent frameworks and four benchmarks, AECG achieves the best score in 11 of 12 framework–benchmark settings and improves over the strongest competing memory method by as much as 10.23 percentage points; removing scope preservation reduces accuracy by up to 16.89 points. AECG thereby reframes multi-agent memory from passive accumulation into auditable reliability governance.Code is available at https://anonymous.4open.science/r/AECG-5A2F

1 INTRODUCTION

Large language model (LLM)-based multi-agent systems increasingly rely on memory to transform execution trajectories into reusable procedural knowledge (Sumers et al., 2023; Wang et al., 2024a; Liu et al., 2024). Prior work has made substantial progress in experience representation, skill abstraction, and memory retrieval, enabling agents to distill strategies from past interactions and reuse them in subsequent tasks (Park et al., 2023; Shinn et al., 2023; Zhao et al., 2024; Wang et al., 2024b; Xu et al., 2026). Yet as memory evolves from trajectory storage into a library of persistently reused skills, a deeper limitation emerges: existing approaches primarily emphasize how knowledge is formed and recalled, leaving its reliability decay under continued reuse comparatively underexplored. Correct retrieval does not imply reliability, nor does it imply that the retrieved knowledge can be safely generalized across different levels of collaboration.

Continued reuse makes this reliability gap increasingly consequential. An isolated error typically affects only a local execution. However, once an outdated, weakly supported, or spuriously successful procedure is consolidated into long-term memory and repeatedly recalled, the error becomes a persistent component of future reasoning. This gives rise to memory pollution and sustained negative transfer across tasks (Zhang et al., 2022; Tang et al., 2026; Wu et al., 2026b; Zhang et al., 2026a).

Multi-agent collaboration amplifies these risks. Its execution spans multiple levels—including team planning, collaborative events, and role-specific actions—whose interdependent decisions form a complex execution structure (Li et al., 2023; Qian et al., 2024; Hong et al., 2024; Du et al., 2023; Zhang et al., 2026b; Han et al., 2025). Reliable multi-agent memory governance must answer three

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.

questions. First, where are the boundaries of valid reuse? Once a local procedure escapes the coordination scope supported by its original evidence, it can trigger scope collapse. Second, when has knowledge degraded? Task-level failures are difficult to attribute to a particular planning, coordination, or execution decision (Kong et al., 2026), while the inertia of historical success can mask a recent reliability decline. Third, how should intervention proceed under bounded computation? Skills occupy different topological positions in the execution graph and therefore have markedly different downstream impact; a system cannot review every suspicious skill with an equal budget.

Together, these questions require multi-agent memory to shift from passive accumulation to continuous lifecycle governance, as illustrated in Figure 1. To this end, we introduce **AECG**, a framework for asymmetric experience consolidation and governance. First, AECG preserves coordination scope throughout experience storage, skill abstraction, and retrieval, ensuring knowledge is reused only within evidence-supported boundaries. Second, it admits success and failure asymmetrically: successful trajectories provide confidence-weighted positive evidence, whereas a failure affects reliability only when its responsible scope and associated skill are consistently attributed. Over the admitted evidence, AECG maintains a cumulative slow belief and a fast belief that emphasizes recent outcomes, revealing degradation otherwise masked by historical success. Finally, it combines degradation with downstream exposure in the execution graph to prioritize high-impact skills under a bounded review budget.

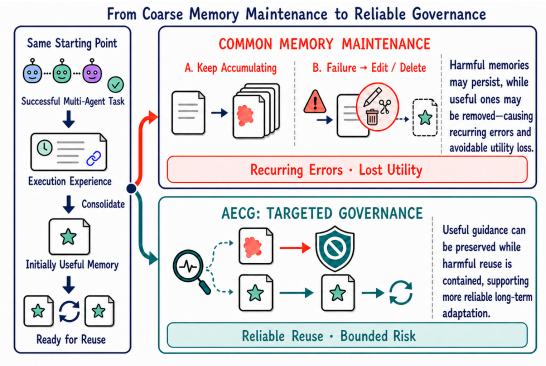

Figure 1: Conceptual comparison of memory maintenance: coarse accumulation or reactive revision risks persistent errors and utility loss, whereas AECG targets governance for reliable reuse.

For each reviewed skill, AECG distinguishes failures caused by overly broad applicability from those caused by a procedural defect and accordingly proposes either context narrowing or procedural repair. A proposal creates a new candidate version rather than overwriting the active skill. The candidate is reactivated only if it mitigates the implicated behavior on failure fixtures while preserving prior supported behavior on historical success fixtures. By doing so, AECG closes the governance loop: it constrains where knowledge is reused, detects when it degrades, prioritizes intervention, and verifies whether a revised skill can safely return to service.

We evaluate AECG across three multi-agent frameworks and four benchmarks. It achieves the best score in 11 of 12 settings, outperforming the strongest memory baseline by up to 10.23 percentage points. Mechanism studies show that removing scope preservation reduces accuracy by up to 16.89 points, dual-timescale beliefs detect degradation faster, and downstream exposure improves high-impact risk prioritization under a fixed review budget.

In summary, our main contributions are as follows:

1. We propose a scope-preserving, asymmetric experience consolidation framework that maintains coordination boundaries during knowledge formation and reuse while controlling negative-evidence admission through conservative failure attribution.

2. We develop a dual-timescale reliability estimator that compares historical and recent beliefs to identify skill degradation masked by accumulated evidence of past success.

3. We develop topology-aware, budgeted governance that combines reliability degradation with exposure and enables safe skill updates through evidence-routed intervention and paired replay.

2 Related Work

Memory and experience reuse in LLM agents. Early agent memory systems retrieve interaction histories, observations, or reflections to inform later decisions (Lewis et al., 2020; Borgeaud et al., 2022; Packer et al., 2023). Generative Agents derives higher-level reflections from experi-

ence streams (Park et al., 2023), while Reflexion and ExpeL retain feedback or transferable lessons from prior trajectories (Shinn et al., 2023; Zhao et al., 2024). More recent systems induce reusable workflows or continually organize memories into structured, revisable representations (Wang et al., 2024b; Xu et al., 2026). This progression makes experience more actionable, but also turns generated abstractions—not only raw observations—into persistent inputs to future reasoning.

Memory in multi-agent systems. Multi-agent memory additionally determines where experience resides, which agent consumes it, and how collaborative traces are represented. G-Memory organizes experience through hierarchical graphs, while LEGOMem separates full-task memories for orchestration from subtask memories for specialized agents (Zhang et al., 2026b; Han et al., 2025). LLMA-Mem studies local, shared, and hybrid memory topologies, and ConMem coordinates useful strategies and failure cues in a relation graph (Wu et al., 2026a; Tan et al., 2026). These designs highlight the importance of representation, placement, and inference-time coordination, but focus primarily on effective construction and retrieval rather than the longitudinal reliability of reused procedures or the propagation of their errors through collaboration.

Procedural consolidation and skill evolution. Procedural memory compresses experience into reusable policies, workflows, or skills (Wei et al., 2022). Voyager refines an executable skill library through environmental feedback (Wang et al., 2023), while MACLA constructs hierarchical procedures, maintains Beta reliability beliefs, and refines their conditions from successes and failures (Forouzandeh et al., 2025). MSCE retains evidence links, applicability boundaries, verification rules, and reliability estimates during skill promotion (Tang et al., 2026); Bayesian-Agent instead treats skills as evidence-bearing hypotheses and uses posterior beliefs to guide their evolution (Wu et al., 2026b). These methods move beyond simple accumulation, yet repeated LLM-based consolidation can still discard evidence or introduce spurious rules (Zhang et al., 2026a). AECG complements this line by jointly preserving coordination scope, admitting failures through conservative attribution, reassessing reliability at two timescales, and governing reuse via downstream exposure.

3 METHOD

3.1 PROBLEM FORMULATION AND METHOD OVERVIEW

We consider a stream of multi-agent tasks $\{x_t\}_{t=1}^{T}$. The execution of each task is formalized as a directed execution graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$, producing a corresponding trace τ_t and a terminal outcome $r_t \in \{0, 1\}$. Each node $v \in \mathcal{V}_t$ represents a planning, coordination, or execution decision, while the edges reflect the conditional dependencies between these decisions. To capture the hierarchical nature of multi-agent collaboration, we map each node to a specific coordination scope:

$$s(v) \in \mathcal{S}_t = \{\textsc{Team}\} \cup \{\textsc{Event}(e)\} \cup \{\textsc{Agent}(e, a)\}. \tag{1}$$

AECG maintains two scope-tagged objects: a procedural memory m grounded in an execution trace and a reusable skill k abstracted from supported procedural experience. Their scope constrains recall, evidence accumulation, abstraction, and comparison.

Persistent knowledge creates two opposing service risks. Reusing an unreliable skill can propagate errors through future executions, whereas restricting a useful skill can withhold guidance. AECG addresses this trade-off through a closed loop: it admits heterogeneous feedback as confidence-weighted evidence, maintains historical and recent Beta-style pseudo-count beliefs, and reviews confidence-qualified skills according to their reliability degradation and distance-discounted downstream exposure. Qualified evidence routes reviewed skill to context narrowing or procedural repair, and paired replay prevents unsafe revisions from activation. Figure 2 summarizes this lifecycle.

3.2 SCOPE-CONSTRAINED EVIDENCE LIFECYCLE

To instantiate this governance loop, AECG must first control how execution experience and feedback enter persistent memory and become reusable knowledge. It does so through scope preservation, asymmetric evidence admission, and evidence-grounded skill formation.

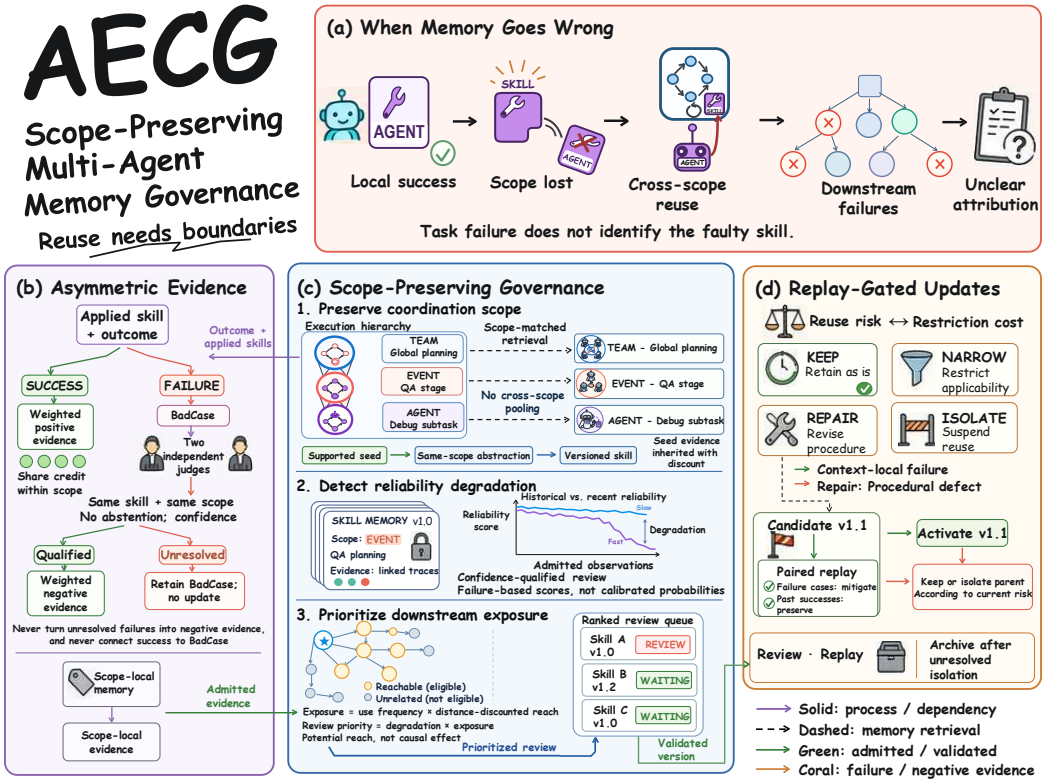

Figure 2: Overview of AECG. Agents retrieve scope-aligned procedural memories and skills. Positive evidence consolidates online, while ambiguous failures await qualified local feedback as BadCases. Scope-preserving promotion yields versioned skills, whose dual-timescale reliability and downstream exposure drive budgeted review and replay-gated governance.

3.2.1 Scope-Preserving Memory Objects

Team-level objects support global planning, event-level objects support a collaborative stage, and agent-level objects support role-specific execution while retaining their source identity. Evidence is not pooled across these levels. When procedural memory m_{seed} is abstracted into skill k, AECG enforces $s(k) = s(m_{seed})$. This constraint is applied during recall and comparison, preventing local experience from becoming system-wide knowledge and causing scope collapse.

3.2.2 Asymmetric Evidence Admission

Scope preservation determines where evidence may be applied; it does not, however, determine whether an observed outcome provides reliable evidence. Task outcomes are inherently asymmetric in this respect: success provides confidence-weighted positive evidence for skills applied in a successful trajectory, whereas task-level failure alone does not identify which persistent object deserves negative evidence. For the u-th use of skill k, let $y_{k,u} \in \{1, 0, \emptyset\}$ denote admitted positive, admitted negative, or unresolved evidence, and let $c_{k,u} \in [0, 1]$ be its confidence weight. Given task outcome r_t, AECG routes evidence as

$$(y_{k,u}, c_{k,u}) = \begin{cases} (1, c^{+}_{k,u}), & r_t = 1 \text{ and } k \text{ was applied}, \\ (\emptyset, 0), & r_t = 0 \text{ and no qualified attribution to } k \text{ is available}, \\ (0, c^{\text{att}}_{k,u}), & r_t = 0 \text{ and } k \text{ passes the attribution gate}. \end{cases} \tag{2}$$

Qualified attribution uses the failed-task execution record, including the applied skills, scopes, and execution nodes. Two LLM judges may abstain and can select only applied skills. Negative evidence is admitted only if both identify the same skill and scope, neither abstains, and their mean confidence meets a predefined attribution threshold; otherwise the observation remains unresolved. Thus, at-

tribution serves as conservative evidence admission rather than ground-truth causal identification. Failed trajectories remain available as BadCases. Appendix B.5 details the protocol.

This admission rule induces, for each skill, the set of observations that may affect its reliability:

$$\mathcal{A}_k = \{u \mid y_{k,u} \in \{0,1\}\}. \tag{3}$$

Only observations in $\mathcal{A}_k$ update the reliability beliefs.

3.2.3 SKILL FORMATION AND EVIDENCE INHERITANCE

Having established which observations count as evidence, AECG next determines when a same-scope procedural memory has accumulated sufficient support to become a reusable skill and how its evidence should be inherited after abstraction. Repeatedly supported procedural memories become candidates for skill formation through a within-scope readiness rule. Its implementation is deferred to Appendix B.1. An LLM abstracts the selected seed into a reusable same-scope skill. Nearby memories may clarify applicability, but only the seed supplies initial evidence.

Let n^+_{seed} and n^-_{seed} denote the numbers of admitted positive and negative observations, $n_{\text{seed}} = n^+_{\text{seed}} + n^-_{\text{seed}}$, and $\widehat{p}_{\text{seed}} = n^+_{\text{seed}}/n_{\text{seed}}$. For $q \in \{S, F\}$, define

$$\mathbf{b}^q_{k,u} = \begin{bmatrix} \alpha^q_{k,u} \\ \beta^q_{k,u} \end{bmatrix}, \qquad \mathbf{b}_0 = \begin{bmatrix} \alpha_0 \\ \beta_0 \end{bmatrix}, \qquad \mathbf{e}(p) = \begin{bmatrix} p \\ 1-p \end{bmatrix}.$$

With initial pseudo-counts $\alpha_0 = \beta_0 = 1$, AECG transfers a fraction $\kappa \in (0, 1]$ of the seed evidence:

$$n^S_{k,0} = \kappa n_{\text{seed}}, \qquad n^F_{k,0} = \min\{W, \kappa n_{\text{seed}}\}, \qquad \mathbf{b}^q_{k,0} = \mathbf{b}_0 + n^q_{k,0}\mathbf{e}(\widehat{p}_{\text{seed}}) \tag{4}$$

The slow state retains all discounted seed evidence, whereas the fast state caps inherited mass at window W. The coefficient κ controls how much seed evidence is inherited by the abstracted skill, limiting the influence of evidence accumulated before abstraction.

3.3 DUAL-TIMESCALE RELIABILITY ESTIMATION

Once a skill is formed with qualified evidence inherited from its seed memory, AECG must translate admitted outcomes into evolving reliability estimates. To separate historical support from recent effectiveness, each active skill version maintains two Beta-style pseudo-count beliefs: one slow and cumulative, and the other fast and discounted. We use these Beta distributions for scoring, rather than as posteriors under a full generative Bayesian model:

$$\theta^S_{k,u} \sim \text{Beta}(\alpha^S_{k,u}, \beta^S_{k,u}), \qquad \theta^F_{k,u} \sim \text{Beta}(\alpha^F_{k,u}, \beta^F_{k,u}). \tag{5}$$

The slow belief provides a historical baseline, whereas the fast belief represents recent effectiveness. For $u \in \mathcal{A}_k$, let $\text{prev}_k(u)$ be the preceding index in $\mathcal{A}_k$, with $\mathbf{b}^q_{k,\text{prev}_k(u)} := \mathbf{b}^q_{k,0}$ if none exists. Setting $\lambda_S = 1$ and $\lambda_F = \lambda < 1$, each admitted observation updates both beliefs as

$$\mathbf{b}^q_{k,u} = \mathbf{b}_0 + \lambda_q\big(\mathbf{b}^q_{k,\text{prev}_k(u)} - \mathbf{b}_0\big) + c_{k,u}\mathbf{e}(y_{k,u}), \quad u \in \mathcal{A}_k. \tag{6}$$

The slow belief accumulates admitted evidence, while the fast belief discounts old evidence toward the initial pseudo-count state (Kim et al., 2026).

The resulting beliefs support complementary assessments of current risk and change over time. Their means and the current failure-risk score are

$$\mu^q_k = \mathbb{E}[\theta^q_k] = \frac{\alpha^q_k}{\alpha^q_k + \beta^q_k}, \quad R^{\text{fail}}_k = 1 - \mu^F_k = \frac{\beta^F_k}{\alpha^F_k + \beta^F_k}. \tag{7}$$

Here R^{fail}_k scores the risk of continuing to use the current version. At the belief-mean level, recent degradation relative to the historical baseline is

$$D_k = \big[\mu^S_k - \mu^F_k\big]_+, \qquad [x]_+ = \max(0, x). \tag{8}$$

Thus, the dual beliefs distinguish a skill that has always been unreliable from one whose recent effectiveness has declined relative to its own history. In implementation, AECG replaces this mean-only signal with a Beta-style belief-sampling estimate R^{drift}_k that discounts discrepancy attributable

to sampling noise. The three quantities serve distinct roles: R_k^{fail} measures current failure tendency, D_k provides an interpretable mean-level measure of recent degradation, and R_k^{drift} is the uncertainty-aware degradation score used for governance. Accordingly, R_k^{fail} is diagnostic, whereas periodic review uses confidence-gated R_k^{drift} and ranks eligible skills jointly with downstream exposure. The estimator and the uncertainty statistic used for audit are provided in Appendix B.2.

3.4 Risk- and Exposure-Aware Memory Governance

The dual-timescale estimator provides confidence-aware evidence of skill degradation, but degradation alone does not determine how limited review capacity should be allocated. Because reviewed skills differ in their potential downstream impact, AECG combines reliability degradation with structural exposure before committing intervention resources. Governance then proceeds in three steps: budgeted triage, evidence-routed intervention, and replay-gated activation.

3.4.1 Exposure-aware budgeted triage

Let $d_{k,u} \in \mathcal{V}_t$ be the node at which use u selected skill k, let $\text{Desc}_t(d) := \{v \in \mathcal{V}_t \setminus \{d\} : d \rightsquigarrow v\}$ denote its unique reachable descendants, and let $\text{dist}_t(d_{k,u}, v)$ be the shortest directed-path distance to a downstream node v. Its observed propagation exposure is

$$e_{k,u}^{\text{obs}} = \sum_{v \in \text{Desc}_t(d_{k,u})} \gamma^{\text{dist}_t(d_{k,u},v)-1}, \qquad 0 < \gamma \leq 1, \tag{9}$$

$$\bar{e}_k = \frac{1}{M_k} \sum_{u \in \mathcal{U}_k^{\text{rec}}} e_{k,u}^{\text{obs}}, \qquad E_k = f_k^{\mathcal{H}} \bar{e}_k. \tag{10}$$

Thus, distance discounts remote descendants, while E_k combines recent mean exposure over $M_k = |\mathcal{U}_k^{\text{rec}}|$ uses with the preceding horizon's selected-use count $f_k^{\mathcal{H}}$. This is potential propagation exposure, not a causal effect estimate.

Exposure alone does not indicate that a skill has degraded; it quantifies only the potential reach of its continued reuse. AECG therefore applies exposure only after reliability evidence has qualified a skill for review. Specifically, the belief-sampling confidence score C_k^{deg} admits a skill to periodic review only when $C_k^{\text{deg}} \geq \eta_{\text{deg}}$ and $R_k^{\text{drift}} > 0$. Eligible skills are ranked by

$$H_k := E_k R_k^{\text{drift}}, \tag{11}$$

and up to the top B_{review} enter the review set. Qualified local failures use an event-driven budget. Confidence estimation, cold-start handling, and budget rules are detailed in Appendix B.3.

3.4.2 Evidence-routed and replay-gated intervention

Once a skill enters review—through periodic exposure-aware triage or qualified event-driven failure handling—the question is how it should be treated. AECG uses qualified failure evidence to determine the intervention type. Context-localized failures yield Narrow, which restricts applicability without changing the procedure; cross-context defects yield Repair, which revises the procedure while preserving scope and intent. If neither condition is supported, no candidate is generated.

Selecting an intervention type is not sufficient for safe deployment: a proposed correction may resolve the observed failure while damaging previously supported behavior. AECG therefore creates a versioned candidate rather than modifying the active skill in place and evaluates it through paired replay. Failure fixtures test whether the implicated behavior is mitigated, while disjoint regression fixtures from prior successful same-scope uses test whether supported behavior is preserved. Let $\widehat{r}_{k,z}^{\text{fix}}$ and $\widehat{r}_{k,z}^{\text{reg}}$ denote candidate success on the two sets, and let $\widehat{r}_{k,\text{parent}}^{\text{reg}}$ denote the active parent's regression success. The replay gate is

$$\text{Gate}(z,k) := \mathbf{1}\{\widehat{r}_{k,z}^{\text{fix}} = 1\} \cdot \mathbf{1}\{\widehat{r}_{k,z}^{\text{reg}} \geq \widehat{r}_{k,\text{parent}}^{\text{reg}} - \epsilon_{\text{reg}}\}, \qquad z \in \{\text{Narrow}, \text{Repair}\}, \tag{12}$$

A candidate is activated only if it mitigates the implicated behavior and preserves prior supported behavior within tolerance ϵ_{reg}. Otherwise, fast-belief confidence routes the skill to continued monitoring or isolation. Action-specific replay and fallback rules are given in Appendix B.3.

Table 1: Performance comparison across frameworks and memory methods, averaged over 3 seeds. Bold indicates the best result, with parentheses showing the absolute percentage-point margin over the runner-up. AECG rows are shaded light green.

MAS	Memory	PDDL	FEVER	SciWorld	MBPP-Plus	Avg.
AutoGen	No-Memory	36.52	71.42	58.62	74.59	63.43
	MacNet-M	31.78	63.91	57.78	76.23	60.97
	MemoryBank	40.12	68.17	56.70	75.41	63.01
	ConMem	40.23	69.84	55.56	73.77	62.66
	G-Memory	38.33	67.61	60.02	77.05	63.87
	AECG	**46.67** (+6.44)	**74.49** (+3.07)	**63.35** (+3.33)	**82.78** (+5.73)	**69.79** (+5.92)
MacNet	No-Memory	45.31	61.22	47.73	75.41	59.95
	MacNet-M	48.44	67.35	60.23	77.87	65.86
	MemoryBank	48.44	68.37	51.14	71.05	61.74
	ConMem	**54.12**	70.41	61.36	73.77	66.57
	G-Memory	49.21	69.39	44.32	71.31	60.62
	AECG	51.56	**71.43** (+1.02)	**71.59** (+10.23)	**79.51** (+1.64)	**70.70** (+4.13)
AgentNet	No-Memory	45.31	59.18	55.68	68.85	59.14
	MacNet-M	51.56	66.33	55.68	72.13	63.17
	MemoryBank	53.13	68.37	62.50	69.67	64.79
	ConMem	50.06	69.39	59.09	71.31	64.26
	G-Memory	48.44	69.22	42.05	63.11	57.21
	AECG	**54.69** (+1.56)	**73.47** (+4.08)	**68.18** (+5.68)	**77.87** (+5.74)	**70.43** (+5.64)

4 EXPERIMENTS

Our experiments test four hypotheses derived from the lifecycle-governance perspective. **RQ1** asks whether AECG improves end-task performance across frameworks and domains; **RQ2** asks whether its governance components are necessary and, whether scope preservation prevents cross-level misuse from inducing scope collapse; **RQ3** asks whether dual-timescale beliefs overcome the inertia of historical success to detect skill degradation promptly; and **RQ4** asks whether downstream exposure improves the allocation of limited review capacity toward consequential degraded skills.

4.1 EXPERIMENTAL SETUP

We evaluate AECG on three MAS frameworks—AutoGen (Wu et al., 2023), MacNet (Qian et al., 2025), and AgentNet (Yang et al., 2026)—and four benchmarks spanning planning, fact verification, interactive reasoning, and code generation: PDDL (Ma et al., 2024), FEVER (Thorne et al., 2018), ScienceWorld (Wang et al., 2022), and MBPP-Plus (Liu et al., 2023). We compare against No-Memory, MacNet-M, MemoryBank (Zhong et al., 2024), ConMem (Tan et al., 2026), and G-Memory (Zhang et al., 2026b). All methods share tasks, prompts, tools, execution limits, and evaluators; execution uses `gpt-4.1-mini`, memory-layer calls use `gpt-4o-mini`, and results average three independently seeded runs. Mechanism tests use AgentNet–ScienceWorld with a shared post-warm-up memory snapshot and matched conditions. Appendix C provides full setup details.

4.2 RQ1: DOES AECG IMPROVE END-TASK PERFORMANCE?

Table 1 shows that AECG achieves the best score in 11 of 12 framework–benchmark settings and ranks first in average performance for three frameworks. Its margin over the strongest memory method is 5.92 on AutoGen, 4.13 on MacNet, and 5.64 on AgentNet. Unlike methods that win only under a particular topology, AECG remains strong across conversational, graph-orchestrated, and decentralized systems. This consistency is the primary evidence for the paper's overall claim: reusable experience benefits from lifecycle governance rather than accumulation and retrieval alone.

The framework-wise pattern also clarifies where governance matters. AECG improves over No-Memory by 6.36 points on AutoGen, 10.75 on MacNet, and 11.29 on AgentNet in average score. The larger gains on graph-structured MacNet and AgentNet are consistent with the motivation that a reused procedure can affect more dependent decisions when collaboration exposes explicit routing

or dependency structure. Importantly, AECG also improves AutoGen, showing that its benefit is not restricted to frameworks with an explicit execution DAG.

Across benchmarks, the largest margin over the runner-up is 10.23 points on MacNet–ScienceWorld, while MBPP-Plus improves consistently across all three frameworks. Both settings involve repeated use of intermediate procedures and therefore stress stale or over-generalized skills. PDDL is the most competitive case: ConMem exceeds AECG by 2.56 points on MacNet, the sole exception among 12 settings. Thus, RQ1 is answered positively at the aggregate level, while also showing that no single memory organization dominates every framework–task pairing. Appendix C.2 provides the full heatmap and additional benchmark-wise analysis.

4.3 RQ2: Are the Governance Components Necessary, and Does Scope Preservation Prevent Scope Collapse?

Table 2 first tests whether the three contributions are functionally necessary rather than merely descriptive additions. Removing scope preservation causes the largest degradation: success drops by 16.89 points on ScienceWorld and 8.34 on MBPP-Plus, establishing scope preservation as the most consequential component in this ablation.

The other ablations connect the aggregate result to Contributions 2 and 3. Without the fast belief, performance drops by 4.76 on ScienceWorld and 6.26 on MBPP-Plus, indicating that cumulative evidence alone is too inert when recent effectiveness changes. Removing exposure from the review priority also degrades both domains despite retaining the same confidence gate and review budget. Table 2 therefore establishes necessity; the remaining RQs test whether the intended mechanisms explain these losses.

Table 2: AgentNet ablations. Parentheses show drops from AECG.

Variant	SciWorld	MBPP+
Full AECG	**68.18**	**77.87**
w/o Scope	51.29 (-16.89)	69.53 (-8.34)
w/o Dual-Time	63.42 (-4.76)	71.61 (-6.26)
w/o Exposure	64.82 (-3.36)	75.29 (-2.58)

Flattening scope causes 36.5% of selected uses to cross their supported coordination scope and reduces success by 16.89 (Table 3). These cross-scope uses have average failed-task exposure of 14.8, indicating that unsupported reuse can reach multiple downstream decisions. Hard matching eliminates cross-scope reuse. Because exposure is a structural reachability measure, this result associates scope collapse with lower success and broader reach without claiming every descendant was causally changed.

Table 3: Scope preservation on AgentNet–ScienceWorld.

Method	Success ↑	Cross-scope ↓	Exposure ↓
Hard-scope	**68.18**	**0.0**	–
Flat-scope	51.29	36.5	14.8

Crucially, the Flat-scope variant reaches only 51.29 on ScienceWorld, falling below the No-Memory baseline of 55.68. Persistent memory therefore becomes counterproductive when knowledge is reused beyond the coordination scope supported by its evidence. Together with the observed 36.5% cross-scope use rate and broader potential exposure, this result supports the central hypothesis behind RQ2: all three governance components contribute, and scope preservation is a necessary safeguard against negative transfer from scope collapse.

4.4 RQ3: Can Dual-Timescale Reliability Overcome Historical Inertia?

We inject controlled skill corruptions to test whether the fast belief makes recent contradictory evidence actionable.

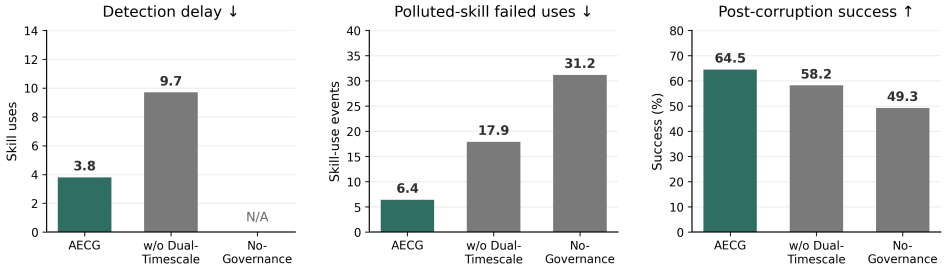

Figure 3: Detection and recovery after controlled skill corruption.

Figure 3 separates rapid detection from the general benefit of having a governance loop. AECG reviews a corrupted skill after 3.8 uses on average, compared with 9.7 for the cumulative-only variant, a 60.8% reduction in delay. Earlier review reduces polluted-skill failed uses from 17.9 to 6.4 and improves post-corruption success from 58.2% to 64.5%.

No-Governance provides a second reference point. It continues to retrieve the corrupted skills but never reviews them, accumulating 31.2 polluted-skill failed uses and reaching only 49.3% success. Thus, detecting drift earlier is useful because it triggers intervention before pollution persists through the remaining task stream. Because AECG and the cumulative-only variant receive the same historical evidence, the reduction in detection delay isolates the role of the fast belief: it prevents recent contradictory outcomes from being overwhelmed by accumulated success. RQ3 therefore confirms that the dual-timescale design counteracts historical inertia, allowing degraded skills to be reviewed earlier and limiting their persistence as memory pollution.

4.5 RQ4: Does Downstream Exposure Improve Triage Under a Fixed Budget?

We compare exposure-aware, risk-only, and random selection under the same confidence gate, corruption set, and review budget. Qualified candidates outnumber available reviews, so the experiment isolates prioritization rather than degradation detection or intervention quality.

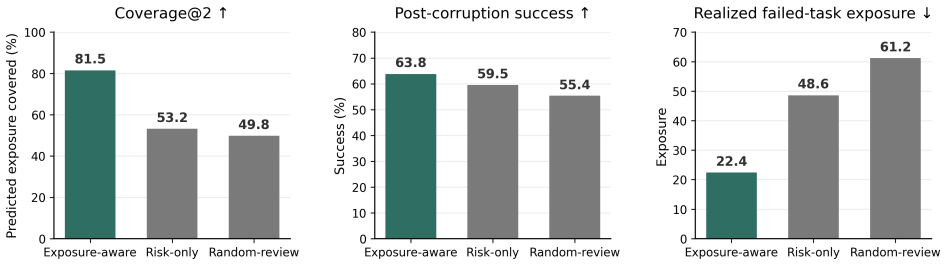

Figure 4: Triage under the same confidence gate and fixed review budget.

Exposure-aware ranking covers 81.5% of polluted-skill predicted exposure in the first two reviews, compared with 53.2% for risk-only and 49.8% for random review (Figure 4). Coverage@2 is only a ranking diagnostic; the primary result is post-corruption success, where exposure-aware triage improves by 4.3 points over risk-only and 8.4 over random review under the same budget.

The realized traces support the same conclusion without reusing the predicted exposure that defines the ranking. Exposure-aware triage reduces failed-task exposure to 22.4, versus 48.6 for risk-only and 61.2 for random review. Because all methods use the same confidence gate, corruption set, intervention rule, and review budget, the performance gain isolates the value of prioritization rather than differences in degradation detection or repair quality. RQ4 supports the governance hypothesis: in a multi-agent execution graph, downstream exposure provides an effective structural signal for directing bounded review capacity toward degraded skills with greater potential propagation impact.

5 Conclusion

We introduce AECG, shifting multi-agent procedural memory from passive accumulation toward lifecycle governance. AECG preserves where knowledge can be reused, detects when it degrades through asymmetric evidence and dual-timescale beliefs, and prioritizes which skills to review using downstream exposure. Evidence-routed intervention and paired replay enable safe skill updates. Across three multi-agent frameworks and four benchmarks, AECG achieves the best performance in 11 of 12 settings. Mechanism studies show that scope preservation mitigates cross-level negative transfer, dual-timescale beliefs overcome historical inertia, and downstream exposure improves review allocation. These results demonstrate that reliable memory requires continuous governance of reuse boundaries, reliability, and potential impact. AECG currently depends on the calibration of localized feedback and treats graph-based exposure as a structural proxy rather than a true causal estimate. Future work will model uncertain responsibility jointly with reliability and replace reachability with intervention-grounded causal influence.

AI USE STATEMENT

Generative AI tools were used to assist with English translation and language polishing, software implementation, and the preparation of selected figures. All AI-assisted text was reviewed and revised by the authors. AI-assisted code was inspected and tested, and AI-assisted figures were checked against the underlying methods and results. The authors take full responsibility for the final content of this work, including all text, code, and figures produced with AI assistance.

REFERENCES

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pp. 2206–2240. PMLR, 2022.

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

Saman Forouzandeh, Wei Peng, Parham Moradi, Xinghuo Yu, and Mahdi Jalili. Learning hierarchical procedural memory for llm agents through bayesian selection and contrastive refinement. *arXiv preprint arXiv:2512.18950*, 2025.

Dongge Han, Camille Couturier, Daniel Madrigal Diaz, Xuchao Zhang, Victor Rühle, and Saravan Rajmohan. Legomem: Modular procedural memory for multi-agent llm systems for workflow automation. *arXiv preprint arXiv:2510.04851*, 2025.

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

Haechan Kim, Soohyun Ryu, Gyouk Chu, Doohyuk Jang, and Eunho Yang. Discounted beta-bernoulli reward estimation for sample-efficient reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2603.18444*, 2026.

Fanqi Kong, Ruijie Zhang, Huaxiao Yin, Guibin Zhang, Xiaofei Zhang, Ziang Chen, Zhaowei Zhang, Xiaoyuan Zhang, Song-Chun Zhu, and Xue Feng. Aegis: Automated error generation and attribution for multi-agent systems. In *International Conference on Learning Representations*, volume 2026, pp. 52994–53028, 2026.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

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

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in neural information processing systems*, 36:21558–21572, 2023.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*, volume 2024, pp. 52989–53046, 2024.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. Agentboard: An analytical evaluation board of multi-turn llm agents. *Advances in neural information processing systems*, 37:74325–74362, 2024.

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

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

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

Chen Qian, Zihao Xie, YiFei Wang, Wei Liu, Kunlun Zhu, Hanchen Xia, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, et al. Scaling large language model-based multi-agent collaboration. In *International Conference on Learning Representations*, volume 2025, pp. 41488–41505, 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.

Theodore R Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L Griffiths. Cognitive architectures for language agents. *arXiv preprint arXiv:2309.02427*, 2023.

Zhixun Tan, Qiang Chen, Tairan Huang, Xiu Su, and Yi Chen. Conmem: Structured memory-guided adaptation in training-free multi-agent systems. *arXiv preprint arXiv:2606.08702*, 2026.

Bo Tang, Yang Zhang, Guomian Zhuang, Wenqiang Wei, Gaoyang Zheng, Lindong Xie, Yanchao Tan, Feiyu Xiong, Qingyu Yang, Edward Chung, et al. From memory to skills: Evidence-grounded co-evolution governance for long-horizon llm agents. *arXiv preprint arXiv:2607.16621*, 2026.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 809–819, 2018.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of computer science*, 18(6):186345, 2024a.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. Scienceworld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11279–11298, 2022.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

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

Shanglin Wu, Yuyang Luo, Yueqing Liang, Kaiwen Shi, Yanfang Ye, Ali Payani, and Kai Shu. Scaling teams or scaling time? memory enabled lifelong learning in llm multi-agent systems. *arXiv preprint arXiv:2604.03295*, 2026a.

Xiaojun Wu, Cehao Yang, Honghao Liu, Xueyuan Lin, Wenjie Zhang, Zhichao Shi, Xuhui Jiang, Chengjin Xu, Jia Li, and Jian Guo. Bayesian-agent: Posterior-guided skill evolution for llm agent harnesses. *arXiv preprint arXiv:2606.08348*, 2026b.

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.

Yingxuan Yang, Huacan Chai, Shuai Shao, Yuanyi Song, Siyuan Qi, Renting Rui, and Weinan Zhang. Agentnet: Decentralized evolutionary coordination for llm-based multi-agent systems. *Advances in Neural Information Processing Systems*, 38:107309–107336, 2026.

Dylan Zhang, Yanshan Lin, Zhengkun Wu, Yihang Sun, Bingxuan Li, Dianqi Li, and Hao Peng. Useful memories become faulty when continuously updated by llms. *arXiv preprint arXiv:2605.12978*, 2026a.

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

Wen Zhang, Lingfei Deng, Lei Zhang, and Dongrui Wu. A survey on negative transfer. *IEEE/CAA Journal of Automatica Sinica*, 10(2):305–329, 2022.

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.

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

APPENDICES

Within this supplementary material, we elaborate on the following aspects:

- Appendix A: Algorithm
- Appendix B: Method Implementation Details
  - Appendix B.1: Evidence Assignment and Skill Promotion
  - Appendix B.2: Beta-Style Belief Drift and Uncertainty (Proposition 1)
  - Appendix B.3: Triage and Action-Specific Validation
  - Appendix B.4: Hyperparameter Selection
  - Appendix B.5: LLM-based Failure Attribution
- Appendix C: Experimental Details
  - Appendix C.1: Benchmarks, Baselines, Frameworks, and Models
  - Appendix C.2: Additional Main-Result Analysis
  - Appendix C.3: Mechanism-Test Protocols
- Appendix D: Prompt Templates
  - Appendix D.1: Skill Abstraction
  - Appendix D.2: Conservative Failure Attribution
  - Appendix D.3: Context Narrowing
  - Appendix D.4: Procedural Repair
  - Appendix D.5: Replay Contract

A ALGORITHM

Algorithm 1 AECG: Asymmetric Experience Consolidation and Governance

Require: $\{x_t\}_{t=1}^{T}$; memory store $\mathcal{M}$; skill registry $\mathcal{K}$; configured parameters and budgets
Ensure: Updated $(\mathcal{M}, \mathcal{K})$

for $t = 1, \ldots, T$ **do**
  // Scope-preserving execution
  Execute x_t with applicable, scope-matched recall at each v; obtain $(\mathcal{G}_t, \tau_t, r_t)$
  // Asymmetric consolidation
  if $r_t = 0$ **then**
    Retain τ_t as a BadCase; compute $G_{k,t}$ by Eq. 24
  end if
  $(y_{k,u}, c_{k,u}) \leftarrow$ Eq. 2
  Update $\mathcal{M}$ from τ_t and admitted scope-local evidence
  for each newly admitted $u \in \mathcal{A}_k$ **do**
    $\mathbf{b}^{q}_{k,u} \leftarrow \mathbf{b}_0 + \lambda_q(\mathbf{b}^{q}_{k,\mathrm{prev}_k(u)} - \mathbf{b}_0) + c_{k,u}\mathbf{e}(y_{k,u}), \quad q \in \{S, F\}$
  end for
  for each m_{seed} passing Eq. 14 **do**
    Abstract k with $s(k) = s(m_{\mathrm{seed}})$
    Register k using Eq. 4; link equivalents without duplicate transfer
  end for
  // Exposure-aware review
  $\mathcal{Q}^* \leftarrow \emptyset$
  if periodic review is due **then**
    $(R_k^{\mathrm{drift}}, C_k^{\mathrm{deg}}) \leftarrow$ Eqs. 15, 21
    $\mathcal{C}_{\mathrm{deg}} \leftarrow \{k \in \mathcal{K} : \ k \text{ active}, \ C_k^{\mathrm{deg}} \geq \eta_{\mathrm{deg}}, \ R_k^{\mathrm{drift}} > 0\}$
    $E_k \leftarrow$ Eq. 10, $\quad H_k \leftarrow E_k R_k^{\mathrm{drift}}$
    $\mathcal{Q}^* \leftarrow \arg \max_{\substack{\mathcal{Q} \subseteq \mathcal{C}_{\mathrm{deg}} \\ |\mathcal{Q}| \leq B_{\mathrm{review}}}} \sum_{k \in \mathcal{Q}} H_k$
  end if
  // Replay-gated governance
  for each distinct k in periodic or budgeted event-driven review **do**
    Propose z_k^{prop} from qualified evidence: NARROW, REPAIR, or no proposal
    For feasible candidates, evaluate $\mathrm{Gate}(z_k^{\mathrm{prop}}, k)$ by Eq. 12
    $z_k^* \leftarrow$ Eq. 23; update the service state in $\mathcal{K}$
  end for
end for
return $(\mathcal{M}, \mathcal{K})$

B METHOD IMPLEMENTATION DETAILS

B.1 EVIDENCE ASSIGNMENT AND SKILL PROMOTION

When execution feedback provides object-level positive confidence, it determines $c^+_{k,u}$. With only task-level success, the $M_{t,s}$ distinct provenance families used in scope s each receive $c^+_{k,u} = 1/M_{t,s}$. A source memory and its derived skill form one family, with effective credit assigned only once. Failed trajectories remain BadCases until local feedback identifies a team-, event-, or agent-level object; unresolved uses produce $(y_{k,u}, c_{k,u}) = (\emptyset, 0)$.

Procedural memory m tracks recent effectiveness with

$$\mathrm{EMA}_{m,j} = \rho\,\mathrm{EMA}_{m,j-1} + (1-\rho)x_{m,j},$$

where j indexes admitted observations, $x_{m,j} = 1$ for positive evidence, and $x_{m,j} = 0$ for qualified negative evidence. Unresolved outcomes do not update the sequence. Define

$$A_m^{\mathrm{ready}} = 1 + n_m^+\,\mathrm{EMA}_m, \qquad B_m^{\mathrm{ready}} = 1 + n_m^-(2 - \mathrm{EMA}_m),$$
$$\mathrm{Ready}(m) = \mu(A_m^{\mathrm{ready}}, B_m^{\mathrm{ready}}) - \sigma(A_m^{\mathrm{ready}}, B_m^{\mathrm{ready}}). \tag{13}$$

Here n_m^+ and n_m^- are the numbers of admitted positive and negative observations. A seed is eligible for promotion only if

$$\mathrm{Ready}(m_{\mathrm{seed}}) > \frac{1}{|\mathcal{P}|} \sum_{m \in \mathcal{P}} \mathrm{Ready}(m) + \delta_{\mathrm{prom}}, \tag{14}$$

where $\mathcal{P}$ contains active memories from the same scope and, at agent level, the same identity; we set $\delta_{\mathrm{prom}} = 0.05$. After promotion, the fast inherited mass in Equation 4 is discounted with $\lambda = 1 - 1/W$. Direct skill-use counts remain zero, and an equivalent existing skill is linked without migrating the same evidence again.

B.2 Beta-Style Belief Drift and Uncertainty

Equation 8 provides an interpretable mean-level degradation signal. To account for Beta-style belief uncertainty in implementation, independently draw paired samples $\theta_{k,j}^S$ and $\theta_{k,j}^F$ from the slow and fast beliefs, and define $\Delta_{k,j} = \theta_{k,j}^S - \theta_{k,j}^F$ and $\bar{\Delta}_k = \mu_k^S - \mu_k^F$. AECG subtracts the centered sampling-noise baseline:

$$R_k^{\mathrm{drift}} = \left[\frac{1}{J} \sum_{j=1}^{J} \left([\Delta_{k,j}]_+ - [\Delta_{k,j} - \bar{\Delta}_k]_+ \right) \right]_+ . \tag{15}$$

The following proposition characterizes the population quantity estimated by Equation 15. It shows that the baseline subtraction is equivalently an expectation under the Beta-style beliefs of clipped positive degradation, rather than a calibrated probability or hypothesis-test statistic.

Proposition 1 (Expected clipped degradation). Let

$$X_k := \theta_k^S - \theta_k^F, \qquad \delta_k := \mathbb{E}[X_k] = \mu_k^S - \mu_k^F,$$

under the independent belief-sampling law used in Equation 15, and define its population counterpart

$$\mathcal{R}_k^{\mathrm{drift}} := \left[\mathbb{E}([X_k]_+ - [X_k - \delta_k]_+) \right]_+ . \tag{16}$$

Then

$$\mathcal{R}_k^{\mathrm{drift}} = \begin{cases} 0, & \delta_k \le 0, \\ \mathbb{E}[\min\{\max\{X_k, 0\}, \delta_k\}], & \delta_k > 0, \end{cases} \tag{17}$$

and, for $\delta_k > 0$,

$$\mathcal{R}_k^{\mathrm{drift}} = \int_0^{\delta_k} \Pr(X_k > t)\, dt. \tag{18}$$

Consequently,

$$0 \le \mathcal{R}_k^{\mathrm{drift}} \le [\delta_k]_+, \qquad \delta_k \Pr(X_k \ge \delta_k) \le \mathcal{R}_k^{\mathrm{drift}} \le \delta_k \Pr(X_k > 0) \quad (\delta_k > 0). \tag{19}$$

Thus, the score is bounded by the belief-mean degradation and attenuates that degradation when belief samples do not consistently support $\theta_k^S > \theta_k^F$. Equivalently, $\mathcal{R}_k^{\mathrm{drift}}/\delta_k$ is the average belief-based exceedance score over thresholds $t \in [0, \delta_k]$.

Proof. Let $h(x) = [x]_+$. If $\delta_k \le 0$, then $X_k - \delta_k \ge X_k$ pointwise. Since h is nondecreasing,

$$h(X_k) - h(X_k - \delta_k) \le 0,$$

and the outer positive part in Equation 16 gives $\mathcal{R}_k^{\mathrm{drift}} = 0$.

Suppose instead that $\delta_k > 0$. A pointwise case analysis gives

$$h(x) - h(x - \delta_k) = \begin{cases} 0, & x \le 0, \\ x, & 0 < x < \delta_k, \\ \delta_k, & x \ge \delta_k, \end{cases} = \min\{\max\{x, 0\}, \delta_k\}.$$

Taking expectations proves Equation 17; the integrand is nonnegative, so the outer positive part is redundant in this case. The layer-cake identity for a nonnegative random variable bounded by δ_k then yields

$$\mathbb{E}[\min\{\max\{X_k, 0\}, \delta_k\}] = \int_0^{\delta_k} \Pr(X_k > t)\, dt,$$

which proves Equation 18. The first pair of bounds in Equation 19 follows immediately from $0 \leq \min\{\max\{X_k, 0\}, \delta_k\} \leq \delta_k$. For $\delta_k > 0$, the pointwise inequalities

$$\delta_k \mathbf{1}\{X_k \geq \delta_k\} \leq \min\{\max\{X_k, 0\}, \delta_k\} \leq \delta_k \mathbf{1}\{X_k > 0\}$$

give the remaining bounds after taking expectations. □

Because each summand in Equation 15 is bounded, the strong law of large numbers implies that its Monte Carlo estimate converges almost surely to $\mathcal{R}_k^{\text{drift}}$ as $J \to \infty$. Moreover, when the sampled belief difference concentrates at a positive value δ_k, dominated convergence gives $\mathcal{R}_k^{\text{drift}} \to \delta_k$, recovering the mean-level degradation in Equation 8. These properties motivate the score as an uncertainty-adjusted degradation magnitude; they do not imply universal superiority over exceedance scores or belief-quantile detectors. We therefore interpret it as a ranking score rather than as a calibrated test statistic.

We use $J = 1024$ with a deterministic seed derived from the belief snapshot. Fast-belief uncertainty is recorded as

$$U_k = \frac{\alpha_k^F \beta_k^F}{(\alpha_k^F + \beta_k^F)^2(\alpha_k^F + \beta_k^F + 1)}. \tag{20}$$

It is retained for audit rather than mixed with hand-set risk weights.

B.3 Triage and Action-Specific Validation

In Equation 9, $\rightsquigarrow$ is a nonempty directed path and converging paths are deduplicated by downstream node. Immediate dependents receive unit weight, while distant decisions receive geometrically smaller weight; at $\gamma = 1$, exposure reduces to descendant counting. This graph-based quantity upper-bounds potential propagation and does not claim that every reachable node was causally changed. Distance discounting also prevents graph depth alone from making high-level skills dominate. We set $\gamma = 0.5$ and use same-scope medians when exposure history is missing.

From the paired samples in Equation 15, the confidence gate is

$$C_k^{\text{deg}} := \frac{1}{J} \sum_{j=1}^{J} \mathbf{1}\{\Delta_{k,j} > 0\}. \tag{21}$$

This Monte Carlo fraction is a confidence score under the Beta-style beliefs, not a calibrated probability of degradation. With $\eta_{\text{deg}} = 0.95$, let $\mathcal{C}_{\text{deg}}$ contain skills satisfying $C_k^{\text{deg}} \geq \eta_{\text{deg}}$ and $R_k^{\text{drift}} > 0$. The periodic review set is

$$\mathcal{Q}^* := \arg \max_{\substack{\mathcal{Q} \subseteq \mathcal{C}_{\text{deg}} \\ |\mathcal{Q}| \leq B_{\text{review}}}} \sum_{k \in \mathcal{Q}} H_k. \tag{22}$$

Thus, review selects the top B_{review} eligible severities. B_{review} is fixed across methods, while qualified local failures use a reserved event-driven budget.

For NARROW, a constrained generator produces a schema-valid predicate over task, tool, event, agent, and input features that excludes the implicated context without changing the procedure. For REPAIR, it preserves scope and intent while revising the procedure or its avoidance rules.

Failure fixtures contain qualified same-scope negative cases; disjoint regression fixtures sample prior successful uses. NARROW withholds predicate-matched uses and runs regression replay outside the excluded region, whereas REPAIR runs the revision on both sets. Missing either set makes the candidate ineligible. We set $\epsilon_{\text{reg}} = 0.05$.

For fallback, define

$$C_k^{\text{unsafe}} := F_{\text{Beta}(\alpha_k^F, \beta_k^F)}(1/2).$$

This fast-belief confidence gate is not a calibrated probability that the skill is unsafe. The final action is

$$z_k^* := \begin{cases} z_k^{\text{prop}}, & z_k^{\text{prop}} \in \{\textsc{Narrow}, \textsc{Repair}\},\ \text{Gate}(z_k^{\text{prop}}, k) = 1, \\ \textsc{Isolate}, & C_k^{\text{unsafe}} \geq \eta_{\text{unsafe}}, \\ \textsc{Keep}, & \text{otherwise}, \end{cases} \tag{23}$$

with $\eta_{\text{unsafe}} = 0.95$. Only candidates that pass paired replay become active versions. If no candidate is generated, either fixture set is missing, or replay fails, the parent version is isolated when $C_k^{\text{unsafe}} \geq \eta_{\text{unsafe}}$; otherwise, it remains active under monitoring. Any unvalidated candidate remains inactive.

B.4 Hyperparameter Selection

To ensure fair evaluation and prevent test-set overfitting, we do not tune hyperparameters on the evaluation benchmarks. Reliability and promotion parameters (κ, W, λ, and δ_{prom}) were selected using a small, independent hold-out development set constructed from a subset of AgentBoard interactive tasks not used in our main evaluation. Governance uses fixed design values $\gamma = 0.5$, $\eta_{\text{deg}} = \eta_{\text{unsafe}} = 0.95$, and $\epsilon_{\text{reg}} = 0.05$; B_{review} is a deployment budget held equal across compared methods rather than a benchmark-tuned accuracy parameter.

Specifically, we perform a grid search on the development set:

- **Evidence Inheritance (κ):** We tested $\kappa \in \{0.3, 0.5, 0.7, 1.0\}$. We set $\kappa = 0.5$ as it optimally balances the preservation of historical confidence with the epistemic uncertainty introduced by LLM-based abstraction.
- **Discount Window (W) and Factor (λ):** We searched $W \in \{5, 10, 20\}$. We selected $W = 10$, which mathematically couples with the fast discount factor $\lambda = 1 - 1/W = 0.9$. This configuration ensures that recent performance degradation (D_k) is statistically detectable within 3-5 consecutive failures, allowing timely governance interventions.
- **Promotion Margin (δ_{prom}):** This operation threshold prevents premature skill abstraction. We empirically set $\delta_{\text{prom}} = 0.05$, which prevents spurious trajectory memorization while ensuring that consistently useful procedures are promoted efficiently.

Importantly, these parameters are kept fixed across all three multi-agent frameworks (AutoGen, MacNet, AgentNet) and all four evaluation benchmarks (PDDL, FEVER, ScienceWorld, MBPP-Plus), demonstrating the generalizability and robustness of the AECG framework without requiring per-task tuning.

B.5 LLM-based Failure Attribution

AECG uses LLM-based attribution only to determine whether a failed execution provides sufficiently qualified negative evidence for a previously applied skill. The attribution component is not treated as a source of ground-truth causal labels.

Applied-skill records. During execution, the memory manager records the skills retrieved, selected, and injected into each agent context, together with their coordination scope, execution node, and downstream nodes. When the agent explicitly reports the skills it used, this report is used as the applied-skill record. For agents without structured skill-use reporting, AECG falls back to a compatibility mode in which injected skills are treated as applied and the record is marked as `compat_assume_injected`. Thus, each applied-skill record is either explicitly reported or conservatively inferred from the injected context.

Failure-case construction. When a task fails, AECG constructs a *BadCase* containing the task specification and input, the final agent output, authoritative environment feedback, the execution trace, and the applied-skill records. Each candidate skill is therefore grounded in an observed execution record rather than introduced by the attribution model.

Dual-judge attribution. Two separately sampled LLM judges independently inspect the BadCase. Each judge outputs a responsible scope, responsible execution node, implicated skill identifier, failure type, confidence score, and an optional abstention decision. Judges are restricted to skills present in the applied-skill record and cannot introduce new skill identifiers.

Let the two judge outputs for failed task t be

$$J_i(t) = (\widehat{s}_i, \widehat{v}_i, \widehat{k}_i, \widehat{f}_i, \widehat{c}_i, a_i), \qquad i \in \{1, 2\},$$

where $\widehat{s}_i$ denotes the predicted scope, $\widehat{v}_i$ the responsible execution node, $\widehat{k}_i$ the implicated skill, $\widehat{f}_i$ the predicted failure type, $\widehat{c}_i \in [0, 1]$ the judge confidence, and $a_i \in \{0, 1\}$ indicates whether the judge abstains.

Attribution gate. A failed observation is admitted as negative evidence for skill k only if all of the following conditions hold: (i) neither judge abstains; (ii) both judges identify the same skill; (iii) both judges identify the same scope; (iv) the identified skill is present in the applied-skill record; and (v) the mean judge confidence is at least 0.70.

Formally, defining

$$\overline{c}_{k,t} = \frac{\widehat{c}_1 + \widehat{c}_2}{2},$$

the attribution gate is

$$G_{k,t} = \mathbb{I}\Big[a_1 = a_2 = 0 \wedge \widehat{k}_1 = \widehat{k}_2 = k \wedge \widehat{s}_1 = \widehat{s}_2 \wedge k \in \mathcal{K}_t^{\text{applied}} \wedge \overline{c}_{k,t} \geq 0.70\Big]. \tag{24}$$

If $G_{k,t} = 1$, the failure contributes $y_{k,u} = 0$ with confidence $c^{\text{att}}_{k,u} = \overline{c}_{k,t}$. Otherwise, the observation is marked unresolved, $y_{k,u} = \emptyset$, and no negative evidence is admitted.

This gate deliberately favors abstention over uncertain attribution. In particular, disagreement between judges, explicit abstention, or insufficient confidence cannot directly decrease the reliability belief of a skill.

C Experimental Details

C.1 Benchmarks, Baselines, Frameworks, and Models

Benchmarks. PDDL, adopted from AgentBoard (Ma et al., 2024), evaluates interactive planning by task success rate. FEVER (Thorne et al., 2018) evaluates evidence-based fact verification by classification accuracy. ScienceWorld (Wang et al., 2022) tests multi-step scientific reasoning and action in a text environment. MBPP-Plus from EvalPlus (Liu et al., 2023) evaluates Python synthesis using augmented hidden tests. Together they span planning, verification, embodied interaction, and code generation.

Baselines. No-Memory removes cross-task persistent memory while retaining native within-task context. MacNet-M retains MacNet's lightweight artifact memory (Qian et al., 2025). MemoryBank uses long-term storage with Ebbinghaus-inspired forgetting and reinforcement (Zhong et al., 2024). ConMem coordinates useful strategies and failure cues as signed cards in a relational graph (Tan et al., 2026). G-Memory organizes insights and condensed collaboration trajectories hierarchically (Zhang et al., 2026b).

Frameworks and controls. AutoGen (Wu et al., 2023) is conversation-centric, MacNet (Qian et al., 2025) organizes collaboration as a directed acyclic graph, and AgentNet (Yang et al., 2026) supports decentralized routing. For every framework–benchmark pair, methods share task instances, prompts, tools, execution limits, evaluators, and memory budgets. Agent execution uses `gpt-4.1-mini`; trajectory consolidation, skill abstraction, memory-side assessment, and intervention proposals use `gpt-4o-mini`. Each configuration is evaluated in three independently seeded runs, and tables report arithmetic means.

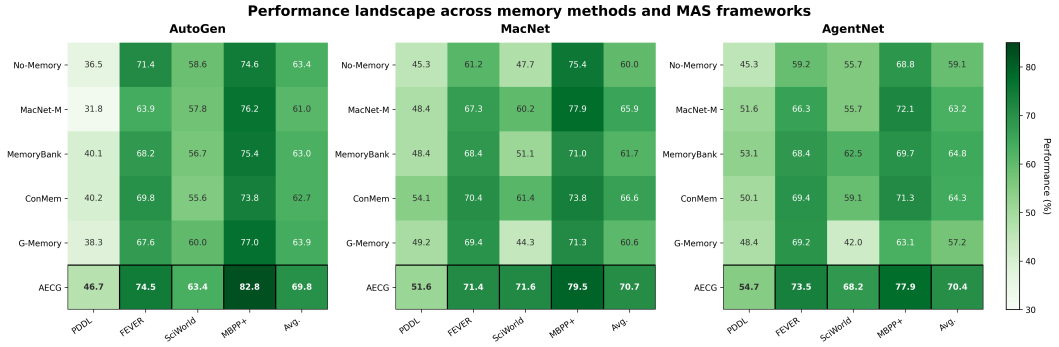

Figure 5: Performance landscape across frameworks, benchmarks, and memory methods. The outlined AECG row corresponds to Table 1.

C.2 Additional Main-Result Analysis

Figure 5 highlights two benchmark-wise patterns. First, the largest improvements occur on ScienceWorld and MBPP-Plus. Under MacNet, AECG raises ScienceWorld accuracy from 47.73 for No-Memory to 71.59 and MBPP-Plus from 75.41 to 79.51; under AgentNet, the corresponding gains are 55.68 to 68.18 and 68.85 to 77.87. These tasks repeatedly reuse intermediate procedures and are especially sensitive to stale or over-generalized skills. Second, PDDL is the most competitive benchmark: ConMem slightly outperforms AECG on MacNet, although AECG remains best on AutoGen and AgentNet. Thus, the overall advantage is not attributable to one framework or task family.

C.3 Mechanism-Test Protocols

Shared protocol. The controlled-corruption and budgeted-triage experiments use AgentNet–ScienceWorld because its execution graph exposes agent-, event-, and team-level decisions and their downstream dependencies. For each seed, the first 60% of the task stream warms memory and the remaining 40% evaluates it. Methods start from the same post-warm-up snapshot and share task order, models, prompts, tools, retrieval limits, and execution budgets.

Controlled corruption. After warm-up, we select four active skills: two with relatively high predicted exposure and two with relatively low exposure. Each has at least three positive observations and a fast-belief mean of at least 0.8. An LLM changes one critical step or applicability condition while preserving scope and principal retrieval cues. On three matched validation tasks, the original skill must succeed at least twice, the corruption must fail at least twice, and the corrupted skill must still be retrieved at least twice. We inject the same validated corruptions into a shared snapshot. Detection delay counts skill uses from the first failed use to first review; polluted-skill failed uses count skill-use events in failed tasks and can exceed the number of tasks.

Scope mechanism. Hard-scope AECG requires source and decision scopes to match, including agent identity for agent-level objects. Flat-scope removes this eligibility constraint while retaining the same top-K and token budget. Cross-scope use rate is the fraction of selected uses whose source and decision scopes differ. Failed-task potential exposure averages their distance-discounted observed exposure in failed tasks and measures potential reach, not causal propagation.

Budgeted triage. We reuse controlled corruption and allow at most one review per round; the number of confidence-qualified candidates exceeds this budget in the first two rounds of every run. We compare $E_k R_k^{\mathrm{drift}}$, R_k^{drift}, and random qualified selection, isolating every selected skill to hold intervention fixed. Coverage@2 is the fraction of total polluted-skill predicted exposure covered by the first two reviews and is a ranking diagnostic. Post-corruption success is the primary outcome, while realized failed-task exposure is computed from post-intervention traces.

D PROMPT TEMPLATES

The following excerpts are abridged versions of the prompts used by AECG. We retain the constraints that determine the behavior of skill formation, failure attribution, evidence-routed intervention, and replay-gated revision. Formatting instructions, few-shot examples, and implementation-specific boilerplate are omitted.

D.1 SKILL ABSTRACTION

SKILL ABSTRACTION PROMPT

Input. A seed procedural memory, its execution trajectory and reliability statistics, up to three auxiliary memories from the same procedural level, and the target skill granularity (`team`, `event`, or `agent`).

Prompt excerpt.

You are the Skill Abstraction Generator in a continual-learning system. Produce a reusable skill from the supplied seed memory.
The seed is the primary evidence. Its procedure must be an abstraction of actions actually present in the seed; do not invent new knowledge, overturn the seed's approach, or import actions that occur only in auxiliary memories. Auxiliary memories may supplement applicability boundaries, risk limits, and stable phrasing, but they are not primary evidence.
Preserve the coordination scope and match the granularity of the skill:

- At the team level, describe how events are decomposed and handed off.
- At the event level, describe how agents divide work and combine outputs.
- At the agent level, describe tool calls, judgments, validation, and fallback behavior for the assigned role.

Remove concrete entities, runtime identifiers, timestamps, and task-specific answers. State reusable applicability conditions and actionable avoidance rules. Every avoidance rule must identify the failure scenario, the resulting risk, and the behavior that should therefore be avoided.
Return exactly the following six fields: `skill_topic`, `applicable_context`, `trigger_condition`, `procedure`, `avoidance`, and `evidence_summary`. The evidence summary must state that the seed is primary evidence and auxiliary memories are supplementary.

D.2 CONSERVATIVE FAILURE ATTRIBUTION

CONSERVATIVE FAILURE ATTRIBUTION PROMPT

Input. The task goal and outcome, the complete execution trace in temporal order, applied-skill records grouped by exact scope, and the full text of each skill that entered the corresponding scope.

Prompt excerpt.

You are one of two independent failure-attribution judges. Determine whether an actually applied skill contributed to the observed failure. Do not infer causality from application, topical similarity, or scope provenance alone.
First identify the observed failure and locate the earliest evidence-supported execution node that deviated from a correct path. For every applied skill, compare its trigger condition, procedure, and avoidance rules with the concrete execution trace. Link a skill to the failure only when a specific clause corresponds to an observed behavior and that behavior participates in the failure chain.
Test competing explanations, including agent noncompliance, planning or coordination errors, bad retrieval, tool or environment failures, insufficient validation, and failures that began before skill application. An applied skill is not proof that the agent followed it.

Recommend at most one exact `scope` and `skill_id`. Abstain when the trace is incomplete, the first deviation cannot be located, multiple candidates are indistinguishable, or a stronger non-skill explanation remains. Prefer conservative abstention over unsupported negative evidence.
Return a structured assessment containing the observed failure, the first deviation, candidate assessments, clause links, alternative explanations, responsibility classification, defect type, failure locality, and an abstention decision. Use only clauses present in the supplied skill text. The two judges are called independently. Negative evidence is admitted only when both judges identify the same applied skill and scope, neither abstains, and their agreement satisfies the confidence gate.

D.3 Context Narrowing

CONTEXT NARROWING PROMPT

Stage A: Failure-Context Predicate
Input. The current skill, attributed bad cases, and the context snapshots recorded at the skill's actual application scope.

Prompt excerpt.

Propose the smallest executable predicate describing the failure context in which the current skill should be excluded. Use only features present in the supplied pre-execution context snapshots and supported by the attributed bad cases. Do not infer features from identifiers, wording, final outcomes, or unsupported domain knowledge.
The predicate uses the implemented matcher semantics: non-empty dimensions are combined with AND; values within a dimension use the specified OR semantics; agent composition requires exact set equality; and feature constraints are conjunctive. Do not use an OR combination that expands beyond the evidence.
Minimize the predicate. Retain a condition only if it distinguishes the failure region, is observable before execution, and can be extracted reliably. Abstain when the context snapshot is missing, the failure regions cannot be represented by one safe predicate, or every non-empty predicate would over-exclude valid uses.
Return either a versioned predicate with its supporting bad cases or an abstention decision. Do not claim that replay has passed.

Stage B: Skill-Boundary Revision
Input. The current skill, attributed bad cases, and the validated failure-context predicate.

Prompt excerpt.

Revise only `applicable_context` and `trigger_condition`. Exclude the predicate-defined failure region while preserving the original skill's supported behavior elsewhere.
Treat the validated predicate as the primary boundary evidence. Use only conditions observable before execution. Do not modify, restate, or supplement the skill's `procedure` or `avoidance`, and do not disable the entire skill unless the evidence covers its full original applicability.
Return the revised applicability and trigger conditions together with a concise rationale identifying what was preserved, what was excluded, and which evidence supports the boundary. The candidate remains inactive until paired replay validates it.

D.4 Procedural Repair

PROCEDURAL REPAIR PROMPT

Input. The current skill, attributed bad cases, the first deviation, clause-level attribution evidence, and relevant counterevidence.

Prompt excerpt.

Revise only `procedure` and `avoidance` to repair the smallest concrete skill defect supported by the attribution evidence.
Distinguish an incorrect instruction, an ambiguous instruction, a missing guardrail, and a wrong avoidance rule. Preserve useful steps, ordering, scope, and intent that are not contradicted by the evidence. Make the revised procedure executable: state branch conditions explicitly, verify critical facts, validate critical outputs, and add stop, retry, fallback, or escalation behavior only when supported by the failure evidence.
Do not change the applicability or trigger conditions. Do not hard-code the answer to one bad case, invent tools or evidence, or introduce an entirely new method. Every avoidance rule must name a concrete dangerous behavior and, where necessary, state the replacement behavior.
Return the revised `procedure`, revised `avoidance`, and an evidence-based rationale. Do not claim that replay has passed; the revision is only a candidate until failure-fixture and regression replay succeed.

D.5 Replay Contract

REPLAY CONTRACT PROMPT

Input. A candidate skill version, its implicated failure fixtures, disjoint historical-success fixtures, and the deterministic replay and governance policy.

Prompt excerpt.

Treat every generated excerpt above as producing a candidate skill version, not as directly modifying an active skill. A candidate is activated only when it mitigates the implicated behavior on failure fixtures and preserves supported behavior on disjoint historical-success fixtures within the regression tolerance.
Otherwise, keep the parent version active, isolate the candidate, or continue monitoring according to the deterministic governance policy. Candidate generation and replay must not be represented as evidence that the revision has passed until both the failure and regression conditions are satisfied.