LEARNING WHERE A LESSON STOPS: REPAIR BOUNDARIES FOR PERSISTENT-MEMORY AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Persistent memory allows language agents to carry lessons across episodes, but a repair learned from one failure can become harmful when reused in a nearby state that was already handled correctly. Existing methods improve what agents store and retrieve, yet the failed trajectory itself gives little evidence about where its correction ceases to be valid. We introduce RANGE, which pairs each repair with an activation support and uses matched successful trajectories as boundary evidence for behavior that should be preserved. This evidence constrains which repair–support lesson is stored and where it may act, and the learned support is enforced again at retrieval. Across ALFWorld, WebShop, and ScienceWorld, RANGE improves post-update success by 5.7–6.0 percentage points over ExpeL while reducing Regression by 45.4–50.4% relative to ExpeL.

1 INTRODUCTION

External persistent memory lets language agents reuse experience across episodes without changing model weights. Agents can store reflections and experiential rules (Shinn et al., 2023; Zhao et al., 2024), reusable workflows (Wang et al., 2025), or structured long-term memories (Packer et al., 2023; Zhong et al., 2024; Xu et al., 2025), and retrieve them when similar situations arise. This supports test-time adaptation over long interactions and has motivated benchmarks for long-horizon memory use (Wu et al., 2025; Hu et al., 2026b). But persistence also allows a local correction to affect later decisions. A lesson learned from one failure may be retrieved in a state where the original behavior was already correct, making memory management itself an important part of agent behavior (Xiong et al., 2026; Hu et al., 2026a). Once lessons persist, the problem is therefore not only what to change, but where that change remains valid.

Existing work addresses parts of this problem. Reflection and failure-learning methods extract corrective signals from unsuccessful interactions (Shinn et al., 2023; Wang et al., 2024; Sun et al., 2026), while experience-based methods turn trajectories into reusable insights or workflows (Zhao et al., 2024; Wang et al., 2025). Other systems improve how persistent memory is stored, updated, retrieved, or conditioned on current state (Packer et al., 2023; Xu et al., 2025; Liang et al., 2026; Wu et al., 2026). More recent approaches make applicability explicit through preconditions, counterevidence, or scope (Qi et al., 2026). These mechanisms can represent where a memory should apply, but not necessarily how that boundary should be inferred from experience. For a failure-derived repair, the failed trajectory shows where behavior needs to change, but says little about nearby states where behavior should be preserved. A useful repair can therefore become harmful when its support extends too far, motivating evidence for where that repair should stop.

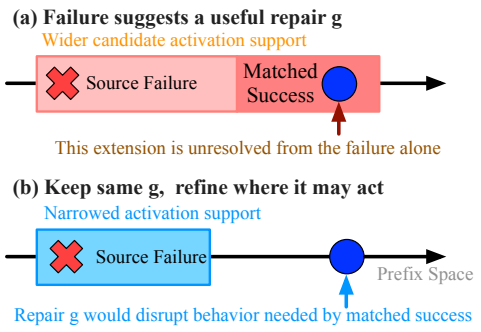

Figure 1: Matched successes provide boundary evidence for narrowing a repair's activation support.

We introduce RANGE (*Repair Activation from Neighboring Goal-Achieving Episodes*) based on a simple asymmetry: a failure shows what needs to change, while a nearby success shows what should be preserved. Suppose a failure suggests a behavioral repair g. If applying g would suppress, replace,

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.

Table 1: Positioning of persistent-memory transfer mechanisms.

Method	Transfer evidence	Applicability	Control
Reflexion (Shinn et al., 2023)	Task feedback	Implicit	Episodic reuse
ExpeL (Zhao et al., 2024)	Success/failure experience	Implicit	Insight update + similarity recall
AWM (Wang et al., 2025)	Demonstrations/experience	Workflow match	Workflow induction + selective reuse
MCMA (Liang et al., 2026)	Task similarity	Abstraction level	Manager-based selective reuse
MemHarness (Wu et al., 2026)	Current state	State-conditioned	Reconstruction before acting
GovMem (Qi et al., 2026)	Support + counterevidence	Assigned scope	Promote/reject/review
RANGE	Matched-success witnesses	Precondition + exception	Accept/reject/narrow + scope-gated retrieval

or materially delay an action needed by a matched success, that successful prefix provides evidence that g should not act there. Figure 1 illustrates this fixed-repair case. RANGE turns such successes into boundary witnesses and learns each repair together with an activation support. During memory update, candidate lessons are accepted or rejected, while overly broad supports can be narrowed using both failure and boundary evidence. At inference, the learned support filters memory before semantic ranking, so a lesson is considered only where its stored scope applies.

We formulate persistent failure learning as joint repair and support estimation, and show why failure evidence alone cannot determine whether a fixed repair should extend to a wider support. RANGE uses matched successful trajectories as boundary evidence during memory admission and enforces the learned support again at retrieval. Across ALFWorld, WebShop, and ScienceWorld, RANGE improves post-update success by 5.7–6.0 percentage points over ExpeL while reducing Regression by 45.4–50.4% relative to ExpeL. Controlled diagnostics and ablations further show the roles of matched-success evidence and scope-gated reuse.

2 Persistent Memory and Repair Boundaries

We consider an interactive language agent whose decisions depend on both the current state and experience retrieved from persistent memory. Because these agents act through sequences of observations and actions (Yao et al., 2023; Schick et al., 2023; Patil et al., 2024), a retrieved lesson can directly alter later behavior. Early work focused on what should be learned from interaction. Reflexion stores verbal feedback, ExpeL distills reusable insights, and AWM turns repeated experience into workflows (Shinn et al., 2023; Zhao et al., 2024; Wang et al., 2025). Other methods learn corrective signals more directly from unsuccessful trajectories (Wang et al., 2024; Sun et al., 2026). Once a correction persists across episodes, the next question is when that correction remains valid.

A second line of work studies how persistent memory is managed and selectively reused. MemGPT, MemoryBank, A-MEM, and Mem0 organize and update persistent memory (Packer et al., 2023; Zhong et al., 2024; Xu et al., 2025; Chhikara et al., 2025), while LongMemEval and MemoryAgentBench study memory over extended interactions (Wu et al., 2025; Hu et al., 2026b). MCMA controls reuse through abstraction, and MemHarness reconstructs retrieved experience from the current state (Liang et al., 2026; Wu et al., 2026). DeltaMem-lite makes corrective patches conditional on preconditions and anti-conditions, while GovMem uses support, counterevidence, and scope when deciding whether a memory should persist (Qi et al., 2026). Table 1 summarizes this progression. The remaining issue for a repair learned from failure is not whether a boundary can be represented, but what evidence should determine that boundary.

Failures and successes provide different information about that boundary. A failure shows that behavior should change near a particular decision, but gives little evidence about nearby states where the original behavior is already correct. A matched success supplies the missing constraint: if the proposed repair would suppress, replace, or materially delay an action needed for that success, the repair should not be active at that prefix. The two trajectories therefore provide different signals. The failure indicates where change is needed, while the success indicates where current behavior should be preserved.

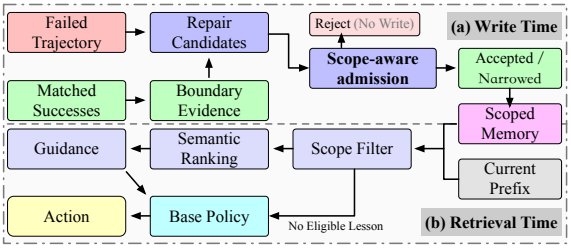

Figure 2: RANGE combines failure and boundary evidence to form scoped lessons at write time, then enforces the learned scope before semantic retrieval.

This distinction is related to counterexample-guided refinement and the balance between stability and plasticity in continual learning (Clarke et al., 2000; Kirkpatrick et al., 2017).

3 RANGE DESIGN

RANGE builds on this observation by learning a repair together with the region in which it may act. The key design choice is to use the same support at both stages of persistent memory: when a lesson is formed and screened at write time, and when it is considered for reuse at retrieval. Figure 2 summarizes this two-stage workflow. The remainder of this section defines the activation support, explains how boundary evidence shapes it, and shows how it is enforced during retrieval.

3.1 REPAIR AND ACTIVATION SUPPORT

Let $\mathcal{H}$ be the space of interaction prefixes, $h \in \mathcal{H}$ the prefix available before the next action, π_0 the frozen base policy, and $y \in \{0, 1\}$ eventual task success. RANGE stores a lesson as

$$\ell = (\phi, b, d, e, c, \rho), \tag{1}$$

where $g_\ell = (b_\ell, d_\ell)$ is the repair payload: b describes behavior to encourage and d behavior to avoid. The Boolean predicates $\phi_\ell, e_\ell : \mathcal{H} \to \{0, 1\}$ define when the lesson may act, with ϕ as the activation condition and e as an exception. The field c stores a compact boundary counterexample that can be used when the candidate is narrowed, and $\rho \in [0, 1]$ is proposal confidence used at retrieval. Writing $\mathbb{I}[\cdot]$ for the indicator function,

$$\mathrm{Act}_\ell(h) = \mathbb{I}[\phi_\ell(h) = 1]\mathbb{I}[e_\ell(h) = 0], \qquad \mathcal{I}(\ell) = \{h \in \mathcal{H} : \mathrm{Act}_\ell(h) = 1\}. \tag{2}$$

Thus g_ℓ specifies what should change, while $\mathcal{I}(\ell)$ specifies where that change may influence the policy. A repair can help at the failed decision and still be harmful elsewhere. Let $\pi_0 \oplus g$ denote the continuation policy when repair g is supplied at prefix h, and define

$$V_\pi(h) = \Pr_\pi(y = 1 \mid h), \qquad \Delta_g(h) = V_{\pi_0 \oplus g}(h) - V_{\pi_0}(h). \tag{3}$$

For a reference distribution μ over $\mathcal{H}$, the value of lesson ℓ with repair g is

$$J_\mu(\ell; g) = \mathbb{E}_{h \sim \mu}[\mathrm{Act}_\ell(h)\Delta_g(h)]\,. \tag{4}$$

Positive $\Delta_g(h)$ means that the repair helps at h, while negative $\Delta_g(h)$ means that it hurts. Let h_f be the source prefix at the failed decision. Once a candidate repair is anchored at h_f, the remaining question is how far its support should extend. To see the ambiguity that remains when only source-side evidence is available, hold the repair g fixed and compare two supports. The narrower lesson ℓ_1 acts on $\mathcal{I}(\ell_1)$, while ℓ_2 also acts on an additional region D. Both include h_f, so the source evidence is compatible with either lesson, but it does not reveal the sign of the repair gain on D. Let $\mathcal{D}_f$ denote the source-side evidence available to a support selector.

Theorem 3.1. *Let ℓ_1 and ℓ_2 share repair g, with measurable supports satisfying $h_f \in \mathcal{I}(\ell_1) \subsetneq \mathcal{I}(\ell_2)$. Let $D = \mathcal{I}(\ell_2) \setminus \mathcal{I}(\ell_1), \mu(D) > 0$. Suppose two measurable repair-gain functions Δ_g^+ and Δ_g^- produce the same source evidence $\mathcal{D}_f$, agree outside D, and satisfy*

$$\Delta_g^+(h) = +\eta, \qquad \Delta_g^-(h) = -\eta, \qquad h \in D,$$

for some $\eta \in (0, 1]$. Let J_μ^+ and J_μ^- denote Equation 4 under these two gain functions. Then any deterministic selector based only on $\mathcal{D}_f$, $\Psi(\mathcal{D}_f) \in \{\ell_1, \ell_2\}$, has, for at least one $\sigma \in \{+, -\}$,

$$\max_{j \in \{1,2\}} J_\mu^\sigma(\ell_j; g) - J_\mu^\sigma(\Psi(\mathcal{D}_f); g) \geq \eta\,\mu(D). \tag{5}$$

Proof. The selector receives the same $\mathcal{D}_f$ in both cases and must therefore make the same choice. If it selects ℓ_1, then under Δ_g^+ the wider lesson ℓ_2 gains $\eta\mu(D)$ on D. If it selects ℓ_2, then under Δ_g^- the narrower lesson ℓ_1 avoids the same loss. □

The missing information is therefore the sign of the repair gain on the additional region D. A nearby matched success can provide a concrete boundary observation: if the wider support would activate g at that prefix and applying g would disrupt behavior needed for the success, the prefix is evidence that the support reaches too far. RANGE uses such successful trajectories as boundary evidence when deciding which repair–support lesson to store.

3.2 Boundary Evidence, Admission, and Refinement

RANGE turns the preceding idea into a single write-time procedure. For a failed episode $f = (x, o_1, a_1, r_1, \ldots, o_H, a_H, r_H, y)$ with horizon H, let x be the task instruction, o_t the observation before action a_t, and r_t the environment feedback. The prefix before action a_t is $h_t = (x, o_1, a_1, r_1, \ldots, o_t)$. Because terminal feedback may arrive after the consequential mistake, RANGE first localizes the decision that should anchor the repair. It builds a nonempty set of candidate spans $\mathcal{S}(f)$ around environment errors, reward changes, action–observation inconsistencies, state transitions, and action boundaries. A controller score $q_{\mathrm{loc}}(s \mid f) \in [0,1]$ selects

$$s_f = [t_1, t_2] = \arg\max_{s \in \mathcal{S}(f)} q_{\mathrm{loc}}(s \mid f), \qquad h_f = h_{t_1}. \tag{6}$$

The localized prefix tells RANGE where the repair must work. It still does not tell the controller where that repair should stop. RANGE next searches the archive $\mathcal{A}^+$ of previously successful trajectories for nearby behavior that should be preserved. For $p \in \mathcal{A}^+$, it computes

$$\mathbf{m}(f,p) = \big(q_{\mathrm{inst}}(f,p), q_{\mathrm{state}}(f,p), q_{\mathrm{act}}(f,p), q_{\mathrm{contrast}}(f,p)\big) \in [0,1]^4, \tag{7}$$

where the four entries measure instruction similarity, local-state similarity, action-schema correspondence, and whether the success provides a useful contrast to the failure. A fixed controller map $Q_{\mathrm{match}} : [0,1]^4 \to [0,1]$ combines these values. With matching cap $k \geq 1$,

$$q_{\mathrm{match}}(f,p) = Q_{\mathrm{match}}(\mathbf{m}(f,p)), \qquad \mathcal{P}_f = \mathrm{TopK}_{p \in \mathcal{A}^+}^{\min(k, |\mathcal{A}^+|)} q_{\mathrm{match}}(f,p), \tag{8}$$

where TopK returns the highest-scoring elements.

A matched success becomes useful boundary evidence only after RANGE identifies what in that trajectory should be protected. For each $p \in \mathcal{P}_f$, the controller identifies a relevant prefix z_p and a set $\mathcal{C}(p)$ of critical actions whose suppression, replacement, or material delay would break the observed successful continuation under the benchmark action budget. These pairs form

$$\mathcal{B}_f = \{(z_p, \mathcal{C}(p)) : p \in \mathcal{P}_f\}. \tag{9}$$

For repair g, the verifier returns a blocking score

$$\zeta(g,p) = Q_{\mathrm{block}}(g; z_p, \mathcal{C}(p)) \in [0,1]. \tag{10}$$

A witness constrains a lesson only when that lesson would actually be active at z_p. Let $n_f = \max(1, |\mathcal{P}_f|)$, with empty sums equal to zero. The compatibility score is

$$K(\ell, \mathcal{P}_f) = 1 - \frac{1}{n_f} \sum_{p \in \mathcal{P}_f} \mathrm{Act}_\ell(z_p)\, \zeta(g_\ell, p). \tag{11}$$

The same verifier provides the binary conflict label used by the Compatibility-violation metric in Section 4. Given the source anchor h_f and boundary witnesses $\mathcal{B}_f$, the controller proposes a finite nonempty set $\mathcal{L}_f$ of repair–support lessons. Each candidate is evaluated on two questions: whether the repair addresses the source failure, and whether its support remains compatible with nearby successful behavior. Source fit is

$$E(\ell, f) = \mathrm{Act}_\ell(h_f)\, Q_E(g_\ell; f, s_f), \tag{12}$$

where $Q_E(g_\ell; f, s_f) \in [0,1]$ scores whether the repair addresses the localized failure. A finite nonempty probe set $\mathcal{X}_f = G_{\mathrm{probe}}(f, s_f)$ tests nearby variants of the same failure pattern,

$$T(\ell, \mathcal{X}_f) = \frac{1}{|\mathcal{X}_f|} \sum_{u \in \mathcal{X}_f} Q_T(\ell; u), \tag{13}$$

with $Q_T(\ell; u) \in [0,1]$. These controller-side probes are separate from the environment-level Transfer cases in Section 4. On the success side, K measures critical-action compatibility. RANGE also estimates broader disruption,

$$\nu(g,p) = Q_R(g; z_p, \mathcal{C}(p)) \in [0,1], \qquad R_{\mathrm{adm}}(\ell, \mathcal{P}_f) = \frac{1}{n_f} \sum_{p \in \mathcal{P}_f} \mathrm{Act}_\ell(z_p) \nu(g_\ell, p), \tag{14}$$

and unsupported breadth,

$$O(\ell) = Q_O(\ell; h_f, \mathcal{B}_f) \in [0, 1]. \tag{15}$$

If no matched success is available, the empty-sum convention gives $K = 1$ and $R_{\mathrm{adm}} = 0$; O still penalizes support that is not justified by the source evidence. Let $\tau_E, \tau_K \in [0, 1]$ be lower-bound thresholds and $\tau_R, \tau_O \in [0, 1]$ upper-bound thresholds. Feasible candidates satisfy

$$\mathcal{F}_f = \{\ell \in \mathcal{L}_f : \; E(\ell, f) \geq \tau_E, \quad K(\ell, \mathcal{P}_f) \geq \tau_K, R_{\mathrm{adm}}(\ell, \mathcal{P}_f) \leq \tau_R, \quad O(\ell) \leq \tau_O\}. \tag{16}$$

The gates define the admissible candidates. Among those candidates, RANGE uses transfer together with the source and boundary scores to choose which lesson to keep:

$$\begin{aligned} U(\ell; f) = \alpha E(\ell, f) + \beta K(\ell, \mathcal{P}_f) + \gamma T(\ell, \mathcal{X}_f) \\ - \delta R_{\mathrm{adm}}(\ell, \mathcal{P}_f) - \epsilon O(\ell), \qquad \alpha, \beta, \gamma, \delta, \epsilon > 0, \end{aligned} \tag{17}$$

and

$$\ell^\star = \arg\max_{\ell \in \mathcal{F}_f} U(\ell; f) \tag{18}$$

when $\mathcal{F}_f \neq \varnothing$. A candidate can fail admission because the repair is wrong, or because the repair is useful but its support is too broad. RANGE refines only the second case. It collects candidates that fit the source but violate at least one boundary condition,

$$\begin{aligned} \mathcal{Q}_f = \{\ell \in \mathcal{L}_f : E(\ell, f) \geq \tau_E, \quad \ell \notin \mathcal{F}_f, \\ K(\ell, \mathcal{P}_f) < \tau_K \; \vee \; R_{\mathrm{adm}}(\ell, \mathcal{P}_f) > \tau_R \; \vee \; O(\ell) > \tau_O\}. \end{aligned} \tag{19}$$

For lessons with the same repair payload, define

$$\ell' \preceq \ell \iff \mathcal{I}(\ell') \subseteq \mathcal{I}(\ell) \; \wedge \; g_{\ell'} = g_\ell, \qquad \ell' \prec \ell \iff \ell' \preceq \ell \; \wedge \; \mathcal{I}(\ell') \subsetneq \mathcal{I}(\ell). \tag{20}$$

When a candidate fits the source but fails the boundary gates, the problem may be its support rather than its repair. RANGE keeps g fixed and first asks whether excluding part of the support is enough. The next proposition shows exactly what such a narrowing changes.

Proposition 3.2. *Let $\ell' \preceq \ell$ share repair content g, with measurable activation sets under the same reference distribution μ. Then*

$$J_\mu(\ell'; g) - J_\mu(\ell; g) = -\int_{\mathcal{I}(\ell) \setminus \mathcal{I}(\ell')} \Delta_g(h) \, d\mu(h). \tag{21}$$

Hence narrowing improves or preserves intervention value exactly when the removed region has non-positive aggregate repair gain.

Proof. Because $\mathcal{I}(\ell') \subseteq \mathcal{I}(\ell)$ and the repair is unchanged, the two terms in Equation 4 cancel on their common support. The difference is the negative contribution of the removed region. □

With g fixed, Equation 21 shows that narrowing changes the objective only through the region that is removed. RANGE cannot evaluate $\Delta_g(h)$ over the full prefix space, but it can evaluate the matched-success witnesses. On those prefixes, $\mathrm{Act}_{\ell'}(z_p) \leq \mathrm{Act}_\ell(z_p)$ and the nonnegative verifier scores imply that narrowing cannot decrease K or increase R_{adm}. These witness-side quantities are the observable signals used to decide where to narrow. If $\mathcal{Q}_f = \varnothing$, nothing is stored. Otherwise RANGE selects $\ell^{(0)} = \arg\max_{\ell \in \mathcal{Q}_f} U(\ell; f)$ and applies one narrowing pass, $\ell^\downarrow = \textsc{Narrow}(\ell^{(0)}, \mathcal{B}_f)$, where NARROW is the controller's support-only refinement operator. A valid revision must satisfy $\ell^\downarrow \prec \ell^{(0)}$: it may strengthen ϕ or add an exception through e, but it cannot change the repair payload. The stored counterexample c may guide this edit. The revised lesson is rescored and stored only if it satisfies Equation 16; otherwise nothing is written for that failure.

3.3 SCOPE-GATED RETRIEVAL

Admission decides which repair–support lessons may persist; retrieval decides whether an admitted lesson may influence the policy at the current prefix. Let $\mathcal{M}$ be the persistent set of admitted lessons. For a future prefix h, RANGE first removes lessons whose support is inactive,

$$\mathcal{M}(h) = \{\ell \in \mathcal{M} : \mathrm{Act}_\ell(h) = 1\} = \{\ell \in \mathcal{M} : h \in \mathcal{I}(\ell)\}. \tag{22}$$

Semantic relevance is evaluated only within this eligible set. With retrieval cap $m \geq 1$, let $q_{\mathrm{rel}}(\ell, h) \in [0,1]$ denote task/state relevance, and let $Q_{\mathrm{ret}} : [0,1]^2 \to [0,1]$ combine it with proposal confidence ρ_ℓ:

$$q_{\mathrm{ret}}(\ell, h) = Q_{\mathrm{ret}}(q_{\mathrm{rel}}(\ell, h), \rho_\ell)\,, \qquad \mathcal{R}(h) = \mathrm{TopK}^{\min(m,|\mathcal{M}(h)|)}_{\ell \in \mathcal{M}(h)}\, q_{\mathrm{ret}}(\ell, h). \tag{23}$$

At most m lessons are selected. Let $\widetilde{\mathcal{R}}(h) \subseteq \mathcal{R}(h)$ denote the selected lessons after removing directly opposing advice for the same decision. Writing a for the next action, RANGE renders the remaining lessons as scoped behavioral advice,

$$\mathcal{G}(h) = \{(\phi_\ell, g_\ell, e_\ell) : \ell \in \widetilde{\mathcal{R}}(h)\}, \qquad \pi_{\mathrm{RANGE}}(a \mid h) = \pi_0(a \mid h, \mathcal{G}(h)). \tag{24}$$

The counterexample c is not shown to the actor, while ρ is used only for retrieval ranking. If $\widetilde{\mathcal{R}}(h) = \varnothing$, the actor uses π_0 without persistent guidance. Thus the support is enforced twice: once when the lesson is admitted and again at the prefix where it might be retrieved.

4 Experiments

4.1 Experimental Setup

• **Tasks and protocol.** We evaluate ALFWorld, WebShop, and ScienceWorld (Shridhar et al., 2021; Yao et al., 2022; Wang et al., 2022), the operating-system and database tasks from AgentBench (Liu et al., 2024), and a tool-use suite drawn approximately evenly from API-Bank, ToolLLM, and τ-bench (Li et al., 2023; Qin et al., 2024; Yao et al., 2025). Each seeded run contains 100 source failures sampled without replacement and processed sequentially as memory updates. For each update, we evaluate pre- and post-update memory states on the source case, three held-out cases with the same localized failure pattern, and three previously solved neighbors, defining PFS, Transfer, and Regression. Evaluation rollouts are read-only. Persistent memory and the successful archive start empty, accumulate within a run, and reset between runs. Preservation cases are disjoint from admission witnesses, and controller probes are disjoint from environment-level Transfer cases. RANGE-Stress contains 25 target–boundary pairs per category and seed. Ablation and sensitivity studies use a separate ALFWorld list with the same 100-update protocol.

• **Baselines and implementation.** We compare with ReAct (Yao et al., 2023), Reflexion (Shinn et al., 2023), and ExpeL (Zhao et al., 2024), together with three diagnostic baselines. Reflection-only stores free-form post-failure advice, Episodic retrieval recalls similar trajectories without success-side admission constraints, and Failure-only keeps RANGE's lesson representation but removes matched-success evidence. Within each suite, methods use the same actor, environment interface, action and termination budgets, seeded source/Transfer/preservation lists, and cross-episode context budget; task failures are not retried. Persistent methods receive at most three retrieved units and 240 tokens; RANGE contributes at most three active lessons, each limited to 80 tokens. GPT-4.1 snapshot `gpt-4.1-2025-04-14` (OpenAI, 2025) is used as the actor in all API experiments and for all RANGE controller calls, with temperature 0. RANGE uses three matched successes, three controller probes, eight candidate lessons per failure, and at most one narrowing pass. The thresholds $(\tau_E, \tau_K, \tau_R, \tau_O) = (0.65, 0.70, 0.20, 0.55)$ and utility weights $(\alpha, \beta, \gamma, \delta, \epsilon) = (0.30, 0.25, 0.20, 0.20, 0.05)$ are selected once on held-out development data and then fixed across all reported suites. Unless noted otherwise, results are mean $\pm$ standard deviation over five seeds, $\{1, 12, 123, 1234, 12345\}$. For actor-capacity experiments, the GPT-4.1 controller is fixed while the actor is replaced by Llama-3.1-70B-Instruct (Grattafiori et al., 2024) or Qwen2.5-7B-Instruct (Yang et al., 2024), both in BF16 with greedy decoding. Llama runs on four NVIDIA A100 80GB GPUs and Qwen on one A100 80GB GPU.

• **Metrics.** For case i, let $y_i^{(0)}, y_i^{(1)} \in [0,1]$ denote normalized outcomes before and after the current memory update, and let $s_i^{(t)} \in \{0,1\}$, $t \in \{0,1\}$, denote benchmark-valid success. *Success* is the mean post-update success over the seven cases associated with each update. For source-pattern cases F and related cases X,

$$\mathrm{PFS}(F) = \frac{1}{|F|} \sum_{i \in F} \left(y_i^{(1)} - y_i^{(0)} \right), \qquad \mathrm{Transfer}(X) = \frac{1}{|X|} \sum_{i \in X} \left(y_i^{(1)} - y_i^{(0)} \right). \tag{25}$$

Table 2: Primary benchmark results.

Benchmark	Method	Success ↑	PFS ↑	Transfer ↑	Reg. ↓
ALFWorld	ReAct	0.512±0.046	0.000±0.000	0.000±0.000	0.000±0.000
	Reflection-only	0.568±0.035	0.139±0.041	0.096±0.026	0.352±0.045
	Reflexion	0.594±0.026	0.181±0.039	0.142±0.022	0.304±0.046
	ExpeL	0.621±0.027	0.218±0.033	0.177±0.026	0.266±0.037
	Episodic retrieval	0.603±0.032	0.196±0.036	0.164±0.030	0.244±0.038
	Failure-only	0.579±0.041	0.231±0.047	0.139±0.034	0.382±0.050
	RANGE	0.681±0.036	0.326±0.056	0.286±0.030	0.132±0.034
WebShop	ReAct	0.454±0.034	0.000±0.000	0.000±0.000	0.000±0.000
	Reflection-only	0.507±0.044	0.132±0.041	0.087±0.022	0.381±0.043
	Reflexion	0.532±0.066	0.171±0.039	0.126±0.010	0.337±0.017
	ExpeL	0.557±0.034	0.204±0.033	0.158±0.034	0.291±0.025
	Episodic retrieval	0.541±0.038	0.183±0.040	0.147±0.026	0.269±0.031
	Failure-only	0.519±0.049	0.219±0.045	0.122±0.030	0.409±0.046
	RANGE	0.617±0.037	0.312±0.045	0.261±0.040	0.151±0.051
ScienceWorld	ReAct	0.386±0.041	0.000±0.000	0.000±0.000	0.000±0.000
	Reflection-only	0.431±0.044	0.118±0.038	0.078±0.026	0.333±0.041
	Reflexion	0.456±0.050	0.156±0.040	0.117±0.022	0.298±0.039
	ExpeL	0.479±0.038	0.190±0.036	0.143±0.029	0.262±0.034
	Episodic retrieval	0.468±0.040	0.174±0.038	0.135±0.031	0.248±0.032
	Failure-only	0.442±0.046	0.201±0.045	0.106±0.033	0.371±0.045
	RANGE	0.536±0.040	0.291±0.048	0.232±0.036	0.143±0.044

For previously solved cases $\mathcal{N}^+$,

$$\mathrm{Reg}(\mathcal{N}^+) = \frac{1}{|\mathcal{N}^+|} \sum_{i \in \mathcal{N}^+} \mathbb{I}\left[s_i^{(0)} = 1 \wedge s_i^{(1)} = 0 \right]. \tag{26}$$

RANGE-Stress additionally reports target gain, scope precision, and compatibility violation. Target gain is the mean $y_i^{(1)} - y_i^{(0)}$ over target cases. If $A_i, G_i \in \{0, 1\}$ indicate whether cross-episode guidance is applied and whether the repair should apply, respectively, scope precision is the fraction of activations with $G_i = 1$ among cases with $A_i = 1$ (defined when at least one activation occurs). Compatibility violation is the fraction of activated boundary cases whose injected memory conflicts with a critical action from the paired success. AgentBench and tool-use experiments also report invalid-action rate and mean normalized task *Score*; *Accepted* is the fraction of source updates that store a lesson. Because ReAct has no cross-episode memory update, its PFS, Transfer, and memory-induced Regression are zero by construction.

4.2 Experimental Results

• **Primary Benchmark Results.** RANGE shifts the recovery–preservation frontier on all three primary benchmarks (Table 2). Relative to ExpeL, the strongest non-RANGE baseline by Success, RANGE improves Success by 5.7–6.0 percentage points and Transfer by 0.089–0.109, while reducing Regression by 45.4–50.4% relative to ExpeL. Failure-only attains relatively strong PFS but also the highest Regression among persistent methods on all three benchmarks, highlighting the cost of learning from the failed trajectory without boundary evidence. Figure 3 shows the same shift toward higher PFS and lower Regression across the three environments.

• **Controlled Failure Generalization.** Failure-only is the closest diagnostic to RANGE: it keeps the lesson representation and the 1+3+3 protocol, but removes matched-success evidence. Restoring matched successes raises PFS from 0.1134 to 0.3657 and Transfer from 0.2050 to 0.3097, while reducing Regression from 0.4307 to 0.1910 (Table 3). The three metrics improve together, showing that success-side evidence shapes a better

Table 3: Controlled failure-generalization results; Δ denotes RANGE minus Failure-only.

Metric	RANGE	Failure-only	Δ
PFS ↑	0.3657	0.1134	+0.2523
Transfer ↑	0.3097	0.2050	+0.1047
Regression ↓	0.1910	0.4307	-0.2397

Table 4: RANGE-Stress boundary diagnostics.

Category	Method	Target gain ↑	Scope prec. ↑	Compat. viol. ↓	Regression ↓
Near-duplicate web	Reflection-only	0.211±0.045	0.512±0.058	0.286±0.044	0.342±0.047
	Reflexion	0.248±0.042	0.583±0.052	0.241±0.040	0.301±0.044
	RANGE	0.364±0.046	0.823±0.036	0.071±0.023	0.124±0.031
Reversible/irreversible	Reflection-only	0.196±0.043	0.498±0.060	0.314±0.045	0.367±0.048
	Reflexion	0.233±0.041	0.561±0.053	0.276±0.042	0.329±0.046
	RANGE	0.342±0.044	0.801±0.039	0.083±0.025	0.142±0.033
Stale/live state	Reflection-only	0.184±0.042	0.476±0.057	0.337±0.048	0.391±0.051
	Reflexion	0.221±0.040	0.548±0.052	0.291±0.043	0.346±0.047
	RANGE	0.331±0.043	0.792±0.040	0.091±0.026	0.151±0.034
Scientific exceptions	Reflection-only	0.172±0.040	0.501±0.059	0.302±0.046	0.358±0.049
	Reflexion	0.209±0.039	0.573±0.054	0.258±0.041	0.314±0.045
	RANGE	0.318±0.042	0.809±0.037	0.076±0.024	0.133±0.032

Table 5: Stateful and tool-use action-validity results.

Suite	Method	Score ↑	PFS ↑	Reg. ↓	Invalid ↓
AgentBench-OS	ReAct	0.438±0.041	0.000±0.000	0.000±0.000	0.083±0.020
	Reflexion	0.507±0.047	0.149±0.041	0.321±0.044	0.118±0.026
	ExpeL	0.536±0.039	0.184±0.038	0.287±0.038	0.102±0.024
	RANGE	0.604±0.035	0.276±0.044	0.139±0.033	0.051±0.017
AgentBench-DB	ReAct	0.462±0.038	0.000±0.000	0.000±0.000	0.071±0.018
	Reflexion	0.529±0.043	0.142±0.039	0.304±0.041	0.097±0.022
	ExpeL	0.553±0.037	0.171±0.035	0.269±0.036	0.086±0.021
	RANGE	0.621±0.034	0.259±0.041	0.127±0.031	0.039±0.015
Tool-use	ReAct	0.488±0.042	0.000±0.000	0.000±0.000	0.102±0.025
	Reflexion	0.557±0.045	0.153±0.040	0.342±0.045	0.136±0.029
	ExpeL	0.586±0.038	0.191±0.037	0.301±0.040	0.114±0.026
	RANGE	0.649±0.036	0.287±0.043	0.151±0.034	0.056±0.018

repair–support hypothesis rather than trading recovery for preservation. In this setting, matched successes affect both which candidate is retained and how broadly it is allowed to act.

• **Boundary Stress Tests.** RANGE-Stress evaluates boundary placement directly by pairing each target with a nearby case where the repair should not apply. Averaged across the four boundary types, RANGE reaches 0.806 scope precision versus 0.566 for Reflexion; compatibility violation falls from 0.267 to 0.080 and Regression from 0.323 to 0.138 (Table 4). Target gain also rises from 0.228 to 0.339, so better preservation comes with stronger improvement on the intended targets. The same pattern appears for near-duplicate items, irreversible actions, stale state, and scientific exceptions.

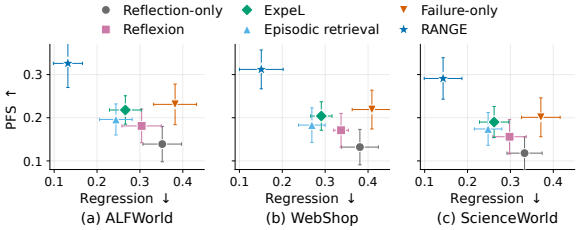

Figure 3: Recovery–preservation frontiers on ALFWorld, WebShop, and ScienceWorld. Points show five-seed means; error bars denote one standard deviation.

• **Action-Level Validity.** AgentBench and the tool-use suite expose scope errors at the action level, before they are reflected in final task success. Relative to ExpeL, RANGE improves normalized Score by 0.063–0.068 while reducing Regression by 0.142–0.150 and invalid-action rate by 0.047–0.058 across AgentBench-OS, AgentBench-DB, and tool use (Table 5). These two error measures operate at different scales: Regression records whether a solved case is lost after the update, whereas invalid-action rate records illegal commands or API calls during execution. Their joint reduction ties the learned support to the concrete action constraints present in successful trajectories.

Table 6: Component ablations.

Variant	Success ↑	PFS ↑	Transfer ↑	Reg. ↓	Scope prec. ↑	Compat. viol. ↓
Full RANGE	0.651±0.039	0.321±0.053	0.274±0.050	0.146±0.029	0.817±0.024	0.071±0.027
No positive traces	0.586±0.057	0.284±0.041	0.183±0.047	0.314±0.033	0.612±0.037	0.219±0.037
No compatibility score	0.603±0.050	0.301±0.044	0.209±0.043	0.286±0.036	0.671±0.039	0.184±0.034
No regression gate	0.602±0.062	0.333±0.043	0.201±0.044	0.352±0.033	0.598±0.054	0.248±0.022
No exceptions	0.621±0.047	0.309±0.046	0.236±0.045	0.218±0.034	0.723±0.036	0.139±0.031
No counterexamples	0.612±0.049	0.298±0.045	0.219±0.044	0.244±0.035	0.704±0.038	0.153±0.032
Unscoped retrieval	0.584±0.052	0.292±0.044	0.192±0.043	0.329±0.038	0.586±0.047	0.231±0.039
Accept all plausible	0.571±0.039	0.347±0.034	0.176±0.032	0.401±0.034	0.544±0.058	0.302±0.045

• **Component Ablations.** Three ablations are especially informative (Table 6). Removing matched positive traces raises Regression from 0.146 to 0.314 and lowers scope precision from 0.817 to 0.612; removing either preservation gate produces a similar loss of boundary quality. Unscoped retrieval raises Regression to 0.329, showing that write-time support must also be enforced at reuse. "Accept all plausible" gives the clearest trade-off: PFS rises to 0.347, but Success falls to 0.571 and Regression reaches 0.401. Together, the ablations separate the roles of boundary evidence, admission control, and scope-gated retrieval. Figure 4 further shows that stronger local repair does not necessarily improve persistent-agent quality; exceptions and stored counterexamples provide smaller refinements.

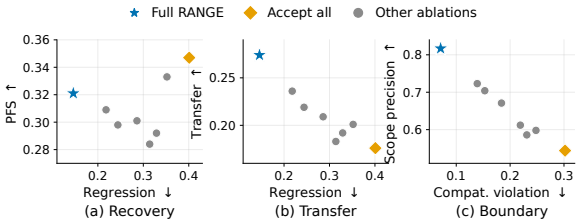

Figure 4: Component ablations across recovery, transfer, and boundary quality.

• **Sensitivity Analysis.** Figure 5 shows a smooth recovery–preservation trade-off rather than a narrow optimum. Increasing matched successes from $k = 0$ to $k = 3$ raises Transfer from 0.183 to 0.274 and lowers Regression from 0.314 to 0.146; beyond $k = 3$, Regression continues to fall, but acceptance declines and PFS no longer improves. A stricter τ_K favors preservation, while a larger τ_R admits more lessons and favors source recovery; the default $k = 3$, $\tau_K = 0.70$, and $\tau_R = 0.20$ lies near the bend of these trends. The results indicate a stable operating trade-off rather than sensitivity to a single setting. Lesson length matters less: increasing the budget from 80 to 120 tokens barely changes PFS or Transfer, but raises Regression.

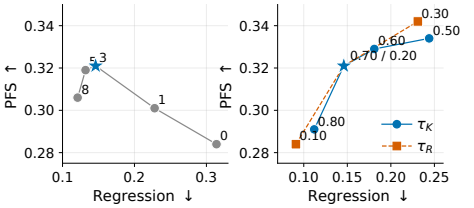

Figure 5: Sensitivity of the recovery–preservation trade-off to matched successes and preservation thresholds; stars are defaults.

• **Actor Capacity Study.** We keep the GPT-4.1 controller fixed and vary only the ALFWorld actor. Relative to Reflexion, the Success gain is 0.087 with GPT-4.1, 0.079 with Llama-3.1-70B-Instruct, and 0.067 with Qwen2.5-7B-Instruct; the Regression reduction remains almost unchanged at 0.170–0.172, and PFS improves with all three actors. The weaker actors convert less of the stored guidance into additional task success, but the preservation benefit changes little over the tested range. The pattern suggests that acting capacity mainly affects how well a lesson is exploited, while boundary control remains comparatively stable under the fixed controller.

5 Conclusions and Limitations

Persistent memory makes local corrections reusable, so their boundaries matter. RANGE uses matched successful trajectories to constrain where a failure-derived repair may act and enforces that scope again at retrieval, improving post-update success while reducing Regression across the evaluated benchmarks. These results suggest that reliable persistent adaptation depends not only on learning useful repairs, but also on controlling where they are allowed to influence future behavior. The learned boundary remains limited by the available evidence: missing or poorly matched successes, or incorrect applicability judgments, can still produce an inaccurate scope.

AI USE STATEMENT

During manuscript preparation, ChatGPT was employed for editorial and organizational support. It was used to refine the presentation of the text, improve the structure and coherence of different sections, assist with literature discovery, and help identify possible inconsistencies in terminology or description. Decisions on content and revisions were made by the authors, rather than adopted directly from the tool. Bibliographic information for relevant references was subsequently verified through Google Scholar before the references were included. Responsibility for the accuracy, interpretation, and final presentation of the work rests with the authors.

REPRODUCIBILITY STATEMENT

The main text specifies the benchmarks, baselines, evaluation protocol, memory budgets, model configurations, controller settings, random seeds, hardware, and metrics. All methods share the same seeded source, Transfer, and preservation cases, and results are reported over five runs unless stated otherwise. The Appendix provides the full update protocol, neighborhood construction, ablation and sensitivity settings, and additional implementation details. We use fixed model snapshots and decoding configurations throughout, and all tasks are drawn from the cited public benchmarks. The reported tables and figures are derived directly from experimental outputs, and the accompanying plotting code reproduces the visualizations used in the paper.

REFERENCES

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready AI agents with scalable long-term memory. In *Proceedings of the 28th European Conference on Artificial Intelligence (ECAI)*, pp. 2993–3000, 2025.

Edmund Clarke, Orna Grumberg, Somesh Jha, Yuan Lu, and Helmut Veith. Counterexample-guided abstraction refinement. In *Proceedings of the 12th International Conference on Computer Aided Verification (CAV)*, pp. 154–169, 2000.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Qisheng Hu, Quanyu Long, and Wenya Wang. When continual learning moves to memory: A study of experience reuse in LLM agents. *arXiv preprint arXiv:2604.27003*, 2026a.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in LLM agents via incremental multi-turn interactions. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pp. 156259–156291, 2026b.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. API-bank: A comprehensive benchmark for tool-augmented LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3102–3116, 2023.

Sirui Liang, Pengfei Cao, Jian Zhao, Wenhao Teng, Xiangwen Liao, Jun Zhao, and Kang Liu. Learning how to remember: A meta-cognitive management method for structured and transferable agent memory. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 30733–30753, 2026.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, et al. AgentBench: Evaluating LLMs as agents. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pp. 52989–53046, 2024.

OpenAI. Introducing GPT-4.1 in the API. OpenAI, April 2025.

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.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive APIs. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 126544–126565, 2024.

Yijiashun Qi, Xiang Xu, and Yuxuan Li. When not to write memory: Governing false promotion from correlated agent traces. In *Proceedings of the 6th International Conference on Machine Learning and Intelligent Systems Engineering (MLISE)*, pp. 228–234, 2026.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, et al. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pp. 9695–9717, 2024.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 68539–68551, 2023.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 8634–8652, 2023.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Xueqiao Sun, Xiaohan Wang, Ludwig Schmidt, Serena Yeung-Levy, and Yuhui Zhang. Learning from failure: Inference-time self-improvement for computer-use agents. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 210–227, 2026.

Renxi Wang, Haonan Li, Xudong Han, Yixuan Zhang, and Timothy Baldwin. Learning from failure: Integrating negative examples when fine-tuning large language models as agents. *arXiv preprint arXiv:2402.11651*, 2024.

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 (EMNLP)*, pp. 11279–11298, 2022.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, pp. 63897–63911, 2025.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

Rong Wu, Daocheng Fu, Licheng Wen, Xuemeng Yang, Shu Zou, Jianbiao Mei, Yuxin Wang, et al. Memharness: Memory is reconstructed, not replayed. *arXiv preprint arXiv:2607.28272*, 2026.

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

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-Mem: Agentic memory for LLM agents. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 17577–17604, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards scalable real-world web interaction with grounded language agents. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 20744–20757, 2022.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik R. Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

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 (AAAI)*, 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 (AAAI)*, volume 38, pp. 19724–19731, 2024.

A Additional Details and Analysis

A.1 Controller and Implementation Details

RANGE follows the same sequence described in Section 3. It first localizes the failed trajectory and sets the source anchor h_f. The successful archive $\mathcal{A}^+$ is then matched to the localized failure, retaining up to $k = 3$ successes $\mathcal{P}_f$ whose critical actions define boundary witnesses $\mathcal{B}_f$. Using h_f and $\mathcal{B}_f$, the controller generates eight candidate lessons $\ell = (\phi, b, d, e, c, \rho)$ and a three-element controller probe set $\mathcal{X}_f$. Each candidate defines a repair $g_\ell = (b_\ell, d_\ell)$ and activation support $\mathcal{I}(\ell)$.

All admission scores lie in $[0, 1]$. E measures fit to the localized source failure and T measures transfer across controller-side variants. On the success side, K measures compatibility with critical actions, R_{adm} estimates disruption of the observed successful continuation, and O penalizes unsupported breadth. The gates use $(\tau_E, \tau_K, \tau_R, \tau_O) = (0.65, 0.70, 0.20, 0.55)$, and utility uses $(\alpha, \beta, \gamma, \delta, \epsilon) = (0.30, 0.25, 0.20, 0.20, 0.05)$. The highest-utility feasible candidate is stored. If no candidate is feasible, RANGE applies at most one support-only refinement to the highest-utility refinable candidate and rescores it once under the same admission rule.

Refinement changes only ϕ and e; the behavioral payload (b, d) remains fixed. The counterexample c can guide this rewrite. At retrieval, support eligibility is checked before semantic ranking. Confidence ρ contributes to ranking, while c is not shown to the actor. Directly conflicting guidance is removed before conditioning the acting model. RANGE retrieves at most three lessons, each capped at 80 tokens, and the actor receives at most three persistent memory units and 240 memory tokens.

A.2 Experimental Protocol

Each seeded run contains 100 source failures sampled without replacement. For update j, we evaluate the source case, three held-out cases with the same localized failure pattern, and three previously solved neighboring cases immediately before and after the memory changes from $\mathcal{M}_{j-1}$ to $\mathcal{M}_j$. Evaluation rollouts are read-only. Preservation cases are disjoint from the successful trajectories used as admission witnesses, and controller probes are disjoint from environment-level Transfer cases. Persistent lessons and the successful archive accumulate across the 100 updates without an additional eviction rule and are reset between runs. All methods reuse the same seeded source, Transfer, and preservation lists. Neighborhoods follow task-relevant state and action structure. ALFWorld uses household goals and local predicates, WebShop uses query intent and requested attributes, and ScienceWorld uses experiment type and target concept. AgentBench and the tool-use suite use command or API schema, mutable state, endpoint or table structure, and action preconditions; the tool-use suite samples approximately equal numbers of source events from API-Bank, ToolLLM, and τ-bench. RANGE-Stress contains 25 target–boundary pairs per category and seed, covering near-duplicate web items, reversible versus irreversible actions, stale versus live state, and scientific

Table 7: Full sensitivity sweep.

Parameter	Value	PFS ↑	Transfer ↑	Reg. ↓	Accepted
k positives	0	0.284±0.041	0.183±0.047	0.314±0.033	0.62±0.06
	1	0.301±0.043	0.226±0.045	0.228±0.035	0.53±0.05
	3	0.321±0.053	0.274±0.050	0.146±0.029	0.41±0.04
	5	0.319±0.050	0.279±0.049	0.132±0.027	0.36±0.04
	8	0.306±0.047	0.273±0.048	0.121±0.026	0.31±0.04
τ_K	0.50	0.334±0.046	0.238±0.044	0.244±0.037	0.57±0.05
	0.60	0.329±0.049	0.261±0.047	0.181±0.033	0.48±0.05
	0.70	0.321±0.053	0.274±0.050	0.146±0.029	0.41±0.04
	0.80	0.291±0.047	0.266±0.048	0.112±0.025	0.29±0.04
τ_R	0.10	0.284±0.046	0.255±0.047	0.091±0.023	0.26±0.03
	0.20	0.321±0.053	0.274±0.050	0.146±0.029	0.41±0.04
	0.30	0.342±0.048	0.263±0.048	0.231±0.036	0.55±0.05
Lesson tokens	40	0.287±0.047	0.239±0.044	0.133±0.028	0.38±0.04
	80	0.321±0.053	0.274±0.050	0.146±0.029	0.41±0.04
	120	0.323±0.050	0.276±0.049	0.171±0.032	0.44±0.04

exceptions. Ablations and sensitivity analysis use a separate ALFWorld case list with the same 100-update protocol. "No positive traces" sets $k = 0$; "No compatibility score" removes the K gate and its utility term; "No regression gate" removes the R_{adm} gate and its utility term; "No exceptions" fixes $e \equiv 0$; "No counterexamples" omits c; "Unscoped retrieval" ranks the full persistent memory without $\mathcal{M}(h)$; and "Accept all plausible" retains source plausibility while removing the preservation gates. All other settings remain fixed.

Strong-API experiments use GPT-4.1 snapshot `gpt-4.1-2025-04-14` for acting and RANGE controller calls with temperature 0. Actor-capacity experiments keep the controller fixed and replace only the actor with Llama-3.1-70B-Instruct or Qwen2.5-7B-Instruct, both in BF16 with greedy decoding. Llama-3.1-70B uses four NVIDIA A100 80GB GPUs with tensor parallelism, while Qwen2.5-7B uses one NVIDIA A100 80GB GPU. The controller configuration is fixed across all reported suites. We use five seeds, $\{1, 12, 123, 1234, 12345\}$. Metrics are computed for each complete seeded run and reported as mean ± standard deviation across seeds, except for the controlled diagnostic, which reports the aggregate mean over the same seeds. Task failures are not retried; transport or parsing failures are repeated only when they do not constitute a new task attempt.

A.3 Additional Experimental Analysis

Table 7 provides the full sensitivity sweep behind Figure 5. Increasing matched successes from $k = 0$ to $k = 3$ raises Transfer from 0.183 to 0.274 and reduces Regression from 0.314 to 0.146; beyond three witnesses, preservation continues to improve, but acceptance falls and PFS no longer increases. The preservation thresholds trace the same trade-off in different directions: a larger τ_K is more conservative, whereas a larger τ_R admits more lessons and favors source recovery. Lesson length has a smaller effect, as increasing the budget from 80 to 120 tokens barely changes PFS or Transfer while increasing Regression. The actor-capacity sweep shows a related separation between using a lesson effectively and preserving neighboring behavior. RANGE improves Success and PFS and lowers Regression with all three actors; the Success gain over Reflexion decreases from 0.087 with GPT-4.1 to 0.067 with Qwen2.5-7B, while the absolute Regression reduction remains between 0.170 and 0.172. Across these settings, recovery depends more strongly on how well the actor can exploit stored guidance, whereas the preservation benefit remains comparatively stable.